IEC 61131-9

Edition 1.0 2013-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Programmable controllers –
Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

Automates programmables –
Partie 9: Interface de communication numérique point à point pour petits capteurs et actionneurs (SDCI)



Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2013 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 la CEI ou du Comité national de la CEI du pays du demandeur.
Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,…).
It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - 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 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) 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 CEI

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

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI 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.

Just Published CEI - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde 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 les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - 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.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

IEC 61131-9

Edition 1.0 2013-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Programmable controllers –
Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

Automates programmables –
Partie 9: Interface de communication numérique point à point pour petits capteurs et actionneurs (SDCI)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

PRICE CODE
CODE PRIX XH

ICS 25.040.40; 35.240.50

ISBN 978-2-8322-1076-5

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

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

CONTENTS

FOREWORD 14
INTRODUCTION 16
1 Scope 18
2 Normative references 18
3 Terms, definitions, symbols, abbreviated terms and conventions 19
3.1 Terms and definitions 19
3.2 Symbols and abbreviated terms 23
3.3 Conventions 25
3.3.1 General 25
3.3.2 Service parameters 25
3.3.3 Service procedures 26
3.3.4 Service attributes 26
3.3.5 Figures 26
3.3.6 Transmission octet order 26
3.3.7 Behavioral descriptions 27
4 Overview of SDCI (IO-Link™) 27
4.1 Purpose of technology 27
4.2 Positioning within the automation hierarchy 28
4.3 Wiring, connectors and power 29
4.4 Communication features of SDCI 29
4.5 Role of a Master 31
4.6 SDCI configuration 32
4.7 Mapping to fieldbuses 32
4.8 Standard structure 32
5 Physical Layer (PL) 33
5.1 General 33
5.1.1 Basics 33
5.1.2 Topology 33
5.2 Physical layer services 34
5.2.1 Overview 34
5.2.2 PL services 35
5.3 Transmitter/Receiver 37
5.3.1 Description method 37
5.3.2 Electrical requirements 37
5.3.3 Timing requirements 41
5.4 Power supply 44
5.4.1 Power supply options 44
5.4.2 Power-on requirements 45
5.5 Medium 45
5.5.1 Connectors 45
5.5.2 Cable 47
6 Standard Input and Output (SIO) 48
7 Data link layer (DL) 48
7.1 General 48
7.2 Data link layer services 50
7.2.1 DL-B services 50

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.2 DL-A services 61
7.3 Data link layer protocol 66
7.3.1 Overview 66
7.3.2 DL-mode handler 67
7.3.3 Message handler 75
7.3.4 Process Data handler 82
7.3.5 On-request Data handler 85
7.3.6 ISDU handler 88
7.3.7 Command handler 92
7.3.8 Event handler 95
8 Application layer (AL) 98
8.1 General 98
8.2 Application layer services 99
8.2.1 AL services within Master and Device 99
8.2.2 AL Services 100
8.3 Application layer protocol 108
8.3.1 Overview 108
8.3.2 On-request Data transfer 108
8.3.3 Event processing 114
8.3.4 Process Data cycles 117
9 System management (SM) 118
9.1 General 118
9.2 System management of the Master 118
9.2.1 Overview 118
9.2.2 SM Master services 120
9.2.3 SM Master protocol 125
9.3 System management of the Device 133
9.3.1 Overview 133
9.3.2 SM Device services 135
9.3.3 SM Device protocol 141
10 Device 148
10.1 Overview 148
10.2 Process Data Exchange (PDE) 149
10.3 Parameter Manager (PM) 149
10.3.1 General 149
10.3.2 Parameter manager state machine 149
10.3.3 Dynamic parameter 151
10.3.4 Single parameter 152
10.3.5 Block parameter 153
10.3.6 Concurrent parameterization access 155
10.3.7 Command handling 155
10.4 Data Storage (DS) 155
10.4.1 General 155
10.4.2 Data Storage state machine 155
10.4.3 DS configuration 157
10.4.4 DS memory space 157
10.4.5 DS Index_List 158
10.4.6 DS parameter availability 158
10.4.7 DS without ISDU 158

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.4.8 DS parameter change indication 158
10.5 Event Dispatcher (ED) 158
10.6 Device features 158
10.6.1 General 158
10.6.2 Device backward compatibility 159
10.6.3 Protocol revision compatibility 159
10.6.4 Factory settings 159
10.6.5 Application reset 159
10.6.6 Device reset 159
10.6.7 Visual SDCI indication 159
10.6.8 Parameter access locking 160
10.6.9 Data Storage locking 160
10.6.10 Device parameter locking 160
10.6.11 Device user interface locking 160
10.6.12 Offset time 160
10.6.13 Data Storage concept 161
10.6.14 Block Parameter 161
10.7 Device design rules and constraints 161
10.7.1 General 161
10.7.2 Process Data 161
10.7.3 Communication loss 161
10.7.4 Direct Parameter 161
10.7.5 ISDU communication channel 162
10.7.6 DeviceID rules related to Device variants 162
10.7.7 Protocol constants 162
10.8 IO Device description (IODD) 163
10.9 Device diagnosis 163
10.9.1 Concepts 163
10.9.2 Events 164
10.9.3 Visual indicators 165
10.10 Device connectivity 166
11 Master 166
11.1 Overview 166
11.1.1 Generic model for the system integration of a Master 166
11.1.2 Structure and services of a Master 166
11.2 Configuration Manager (CM) 169
11.2.1 General 169
11.2.2 Configuration parameter 171
11.2.3 State machine of the Configuration Manager 173
11.3 Data Storage (DS) 175
11.3.1 Overview 175
11.3.2 DS data object 175
11.3.3 DS state machine 175
11.3.4 Parameter selection for Data Storage 181
11.4 On-Request Data exchange (ODE) 181
11.5 Diagnosis Unit (DU) 182
11.6 PD Exchange (PDE) 183
11.6.1 General 183
11.6.2 Process Data mapping 183

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.6.3 Process Data invalid/valid qualifier status 184
11.7 Port and Device configuration tool (PDCT) 185
11.7.1 General 185
11.7.2 Basic layout examples 185
11.8 Gateway application 186
11.8.1 General 186
11.8.2 Changing Device configuration including Data Storage 186
11.8.3 Parameter server and recipe control 186
11.8.4 Anonymous parameters 186
11.8.5 Virtual port mode DIwithSDCI 187
Annex A (normative) Codings, timing constraints, and errors 190
Annex B (normative) Parameter and commands 211
Annex C (normative) ErrorTypes (ISDU errors) 228
Annex D (normative) EventCodes (diagnosis information) 233
Annex E (normative) Data types 236
Annex F (normative) Structure of the Data Storage data object 247
Annex G (normative) Master and Device conformity 248
Annex H (informative) Residual error probabilities 254
Annex I (informative) Example sequence of an ISDU transmission 256
Annex J (informative) Recommended methods for detecting parameter changes 258
Bibliography 259

Figure 1 – Example of a confirmed service 26
Figure 2 – Memory storage and transmission order for WORD based data types 27
Figure 3 – SDCI compatibility with IEC 61131-2 27
Figure 4 – Domain of the SDCI technology within the automation hierarchy 28
Figure 5 – Generic Device model for SDCI (Master's view) 29
Figure 6 – Relationship between nature of data and transmission types 30
Figure 7 – Object transfer at the application layer level (AL) 31
Figure 8 – Logical structure of Master and Device 32
Figure 9 – Three wire connection system 33
Figure 10 – Topology of SDCI 34
Figure 11 – Physical layer (Master) 34
Figure 12 – Physical layer (Device) 35
Figure 13 – Line driver reference schematics 37
Figure 14 – Receiver reference schematics 37
Figure 15 – Reference schematics for SDCI 3-wire connection system 38
Figure 16 – Voltage level definitions 38
Figure 17 – Switching thresholds 39
Figure 18 – Format of an SDCI UART frame 41
Figure 19 – Eye diagram for the 'H' and 'L' detection 42
Figure 20 – Eye diagram for the correct detection of a UART frame 42
Figure 21 – Wake-up request 44
Figure 22 – Power-on timing for Power1 45

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 23 – Pin layout front view 46
Figure 24 – Class A and B port definitions 47
Figure 25 – Reference schematic for effective line capacitance and loop resistance 47
Figure 26 – Structure and services of the data link layer (Master) 49
Figure 27 – Structure and services of the data link layer (Device) 49
Figure 28 – State machines of the data link layer 67
Figure 29 – Example of an attempt to establish communication 67
Figure 30 – Failed attempt to establish communication 68
Figure 31 – Retry strategy to establish communication 68
Figure 32 – Fallback procedure 69
Figure 33 – State machine of the Master DL-mode handler 70
Figure 34 – Submachine 1 to establish communication 71
Figure 35 – State machine of the Device DL-mode handler 73
Figure 36 – SDCI message sequences 75
Figure 37 – Overview of M-sequence types 76
Figure 38 – State machine of the Master message handler 77
Figure 39 – Submachine "Response 3" of the message handler 78
Figure 40 – Submachine "Response 8" of the message handler 78
Figure 41 – Submachine "Response 15" of the message handler 78
Figure 42 – State machine of the Device message handler 81
Figure 43 – Interleave mode for the segmented transmission of Process Data 83
Figure 44 – State machine of the Master Process Data handler 83
Figure 45 – State machine of the Device Process Data handler 85
Figure 46 – State machine of the Master On-request Data handler 86
Figure 47 – State machine of the Device On-request Data handler 87
Figure 48 – Structure of the ISDU 88
Figure 49 – State machine of the Master ISDU handler 90
Figure 50 – State machine of the Device ISDU handler 91
Figure 51 – State machine of the Master command handler 93
Figure 52 – State machine of the Device command handler 94
Figure 53 – State machine of the Master Event handler 96
Figure 54 – State machine of the Device Event handler 97
Figure 55 – Structure and services of the application layer (Master) 98
Figure 56 – Structure and services of the application layer (Device) 99
Figure 57 – OD state machine of the Master AL 109
Figure 58 – OD state machine of the Device AL 110
Figure 59 – Sequence diagram for the transmission of On-request Data 112
Figure 60 – Sequence diagram for On-request Data in case of errors 113
Figure 61 – Sequence diagram for On-request Data in case of timeout 113
Figure 62 – Event state machine of the Master AL 114
Figure 63 – Event state machine of the Device AL 115
Figure 64 – Single Event scheduling 116
Figure 65 – Sequence diagram for output Process Data 117

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 66 – Sequence diagram for input Process Data 118
Figure 67 – Structure and services of the Master system management 119
Figure 68 – Sequence chart of the use case "port x setup" 120
Figure 69 – Main state machine of the Master system management 126
Figure 70 – SM Master submachine CheckCompatibility_1 128
Figure 71 – Activity for state "CheckVxy" 130
Figure 72 – Activity for state "CheckCompV10" 130
Figure 73 – Activity for state "CheckComp" 131
Figure 74 – Activity (write parameter) in state "RestartDevice" 131
Figure 75 – SM Master submachine CheckSerNum_3 132
Figure 76 – Activity (check SerialNumber) for state CheckSerNum_3 133
Figure 77 – Structure and services of the system management (Device) 134
Figure 78 – Sequence chart of the use case "INACTIVE – SIO – SDCI – SIO" 135
Figure 79 – State machine of the Device system management 142
Figure 80 – Sequence chart of a regular Device startup 145
Figure 81 – Sequence chart of a Device startup in compatibility mode 146
Figure 82 – Sequence chart of a Device startup when compatibility fails 147
Figure 83 – Structure and services of a Device 148
Figure 84 – The Parameter Manager (PM) state machine 150
Figure 85 – Positive and negative parameter checking result 152
Figure 86 – Positive block parameter download with Data Storage request 153
Figure 87 – Negative block parameter download 154
Figure 88 – The Data Storage (DS) state machine 156
Figure 89 – Data Storage request message sequence 157
Figure 90 – Cycle timing 160
Figure 91 – Event flow in case of successive errors 165
Figure 92 – Device LED indicator timing 165
Figure 93 – Generic relationship of SDCI technology and fieldbus technology 166
Figure 94 – Structure and services of a Master 168
Figure 95 – Relationship of the common Master applications 168
Figure 96 – Sequence diagram of configuration manager actions 170
Figure 97 – Ports in MessageSync mode 171
Figure 98 – State machine of the Configuration Manager 173
Figure 99 – Main state machine of the Data Storage mechanism 175
Figure 100 – Submachine "UpDownload_2" of the Data Storage mechanism 176
Figure 101 – Data Storage submachine "Upload_7" 177
Figure 102 – Data Storage upload sequence diagram 177
Figure 103 – Data Storage submachine "Download_10" 178
Figure 104 – Data Storage download sequence diagram 178
Figure 105 – State machine of the On-request Data Exchange 181
Figure 106 – System overview of SDCI diagnosis information propagation via Events 183
Figure 107 – Process Data mapping from ports to the gateway data stream 184
Figure 108 – Propagation of PD qualifier status between Master and Device 184

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 109 – Example 1 of a PDCT display layout 185
Figure 110 – Example 2 of a PDCT display layout 186
Figure 111 – Alternative Device configuration 187
Figure 112 – Virtual port mode "DIwithSDCI" 188
Figure A.1 – M-sequence control 190
Figure A.2 – Checksum/M-sequence type octet 191
Figure A.3 – Checksum/status octet 192
Figure A.4 – Principle of the checksum calculation and compression 193
Figure A.5 – M-sequence TYPE_0 194
Figure A.6 – M-sequence TYPE_1_1 194
Figure A.7 – M-sequence TYPE_1_2 195
Figure A.8 – M-sequence TYPE_1_V 195
Figure A.9 – M-sequence TYPE_2_1 196
Figure A.10 – M-sequence TYPE_2_2 196
Figure A.11 – M-sequence TYPE_2_3 196
Figure A.12 – M-sequence TYPE_2_4 197
Figure A.13 – M-sequence TYPE_2_5 197
Figure A.14 – M-sequence TYPE_2_6 197
Figure A.15 – M-sequence TYPE_2_V 198
Figure A.16 – M-sequence timing 201
Figure A.17 – I-Service octet 203
Figure A.18 – Check of ISDU integrity via CHKPDU 205
Figure A.19 – Examples of request formats for ISDUs 206
Figure A.20 – Examples of response ISDUs 206
Figure A.21 – Examples of read and write request ISDUs 207
Figure A.22 – Structure of StatusCode type 1 208
Figure A.23 – Structure of StatusCode type 2 208
Figure A.24 – Indication of activated Events 209
Figure A.25 – Structure of the EventQualifier 209
Figure B.1 – Classification and mapping of Direct Parameters 211
Figure B.2 – MinCycleTime 213
Figure B.3 – M-sequence Capability 214
Figure B.4 – RevisionID 215
Figure B.5 – ProcessDataIn 215
Figure B.6 – Index space for ISDU data objects 217
Figure B.7 – Structure of the Offset Time 226
Figure E.1 – Coding examples of UIntegerT 237
Figure E.2 – Coding examples of IntegerT 239
Figure E.3 – Singular access of StringT 240
Figure E.4 – Coding example of OctetStringT 241
Figure E.5 – Definition of TimeT 241
Figure E.6 – Example of an ArrayT data structure 243
Figure E.7 – Example 2 of a RecordT structure 245

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure E.8 – Example 3 of a RecordT structure 245
Figure E.9 – Write requests for example 3 246
Figure G.1 – Test setup for electrostatic discharge (Master) 250
Figure G.2 – Test setup for RF electromagnetic field (Master) 250
Figure G.3 – Test setup for fast transients (Master) 251
Figure G.4 – Test setup for RF common mode (Master) 251
Figure G.5 – Test setup for electrostatic discharges (Device) 252
Figure G.6 – Test setup for RF electromagnetic field (Device) 252
Figure G.7 – Test setup for fast transients (Device) 252
Figure G.8 – Test setup for RF common mode (Device) 253
Figure H.1 – Residual error probability for the SDCI data integrity mechanism 254
Figure I.1 – Example for ISDU transmissions *(1 of 2)* 256

Table 1 – Service assignments of Master and Device 35
Table 2 – PL_SetMode 35
Table 3 – PL_WakeUp 36
Table 4 – PL_Transfer 36
Table 5 – Electric characteristics of a receiver 39
Table 6 – Electric characteristics of a Master port 39
Table 7 – Electric characteristics of a Device 40
Table 8 – Dynamic characteristics of the transmission 43
Table 9 – Wake-up request characteristics 44
Table 10 – Power-on timing 45
Table 11 – Pin assignments 46
Table 12 – Cable characteristics 47
Table 13 – Cable conductor assignments 48
Table 14 – Service assignments within Master and Device 50
Table 15 – DL_ReadParam 51
Table 16 – DL_WriteParam 51
Table 17 – DL_Read 52
Table 18 – DL_Write 53
Table 19 – DL_ISDUTransport 54
Table 20 – DL_ISDUAbort 55
Table 21 – DL_PDOutputUpdate 55
Table 22 – DL_PDOutputTransport 56
Table 23 – DL_PDInputUpdate 57
Table 24 – DL_PDInputTransport 57
Table 25 – DL_PDCycle 58
Table 26 – DL_SetMode 58
Table 27 – DL_Mode 59
Table 28 – DL_Event 60
Table 29 – DL_EventConf 60
Table 30 – DL_EventTrigger 61

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 31 – DL_Control 61
Table 32 – DL-A services within Master and Device 62
Table 33 – OD 62
Table 34 – PD 63
Table 35 – EventFlag 64
Table 36 – PDInStatus 65
Table 37 – MHInfo 65
Table 38 – ODTrig 66
Table 39 – PDTrig 66
Table 40 – Wake-up procedure and retry characteristics 69
Table 41 – Fallback timing characteristics 70
Table 42 – State transition tables of the Master DL-mode handler 71
Table 43 – State transition tables of the Device DL-mode handler 74
Table 44 – State transition table of the Master message handler 79
Table 45 – State transition tables of the Device message handler 82
Table 46 – State transition tables of the Master Process Data handler 84
Table 47 – State transition tables of the Device Process Data handler 85
Table 48 – State transition tables of the Master On-request Data handler 86
Table 49 – State transition tables of the Device On-request Data handler 88
Table 50 – FlowCTRL definitions 89
Table 51 – State transition tables of the Master ISDU handler 90
Table 52 – State transition tables of the Device ISDU handler 92
Table 53 – Control codes 93
Table 54 – State transition tables of the Master command handler 93
Table 55 – State transition tables of the Device command handler 94
Table 56 – Event memory 95
Table 57 – State transition tables of the Master Event handler 96
Table 58 – State transition tables of the Device Event handler 97
Table 59 – AL services within Master and Device 99
Table 60 – AL_Read 100
Table 61 – AL_Write 101
Table 62 – AL_Abort 102
Table 63 – AL_GetInput 102
Table 64 – AL_NewInput 103
Table 65 – AL_SetInput 104
Table 66 – AL_PDCycle 104
Table 67 – AL_GetOutput 105
Table 68 – AL_NewOutput 105
Table 69 – AL_SetOutput 106
Table 70 – AL_Event 107
Table 71 – AL_Control 108
Table 72 – States and transitions for the OD state machine of the Master AL 109
Table 73 – States and transitions for the OD state machine of the Device AL 111

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 74 – State and transitions of the Event state machine of the Master AL 114
Table 75 – State and transitions of the Event state machine of the Device AL 115
Table 76 – SM services within the Master 121
Table 77 – SM_SetPortConfig 121
Table 78 – Definition of the InspectionLevel (IL) 122
Table 79 – Definitions of the Target Modes 123
Table 80 – SM_GetPortConfig 123
Table 81 – SM_PortMode 124
Table 82 – SM_Operate 125
Table 83 – State transition tables of the Master system management 127
Table 84 – State transition tables of the Master submachine CheckCompatibility_1 128
Table 85 – State transition tables of the Master submachine CheckSerNum_3 132
Table 86 – SM services within the Device 136
Table 87 – SM_SetDeviceCom 136
Table 88 – SM_GetDeviceCom 137
Table 89 – SM_SetDeviceIdent 138
Table 90 – SM_GetDeviceIdent 139
Table 91 – SM_SetDeviceMode 140
Table 92 – SM_DeviceMode 141
Table 93 – State transition tables of the Device system management 142
Table 94 – State transition tables of the PM state machine 150
Table 95 – Definitions of parameter checks 152
Table 96 – State transition table of the Data Storage state machine 156
Table 97 – Overview of the protocol constants for Devices 162
Table 98 – Classification of Device diagnosis incidents 164
Table 99 – Timing for LED indicators 165
Table 100 – Internal variables and Events to control the common Master applications 169
Table 101 – State transition tables of the Configuration Manager 174
Table 102 – States and transitions of the Data Storage state machines 179
Table 103 – State transition table of the ODE state machine 181
Table 104 – State transitions of the state machine "DIwithSDCI" 188
Table A.1 – Values of communication channel 190
Table A.2 – Values of R/W 191
Table A.3 – Values of M-sequence types 191
Table A.4 – Data types for user data 192
Table A.5 – Values of PD status 192
Table A.6 – Values of the Event flag 193
Table A.7 – M-sequence types for the STARTUP mode 198
Table A.8 – M-sequence types for the PREOPERATE mode 198
Table A.9 – M-sequence types for the OPERATE mode (legacy protocol) 199
Table A.10 – M-sequence types for the OPERATE mode 199
Table A.11 – Recommended MinCycleTimes 201
Table A.12 – Definition of the nibble "I-Service" 203

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.13 – ISDU syntax 204
Table A.14 – Definition of nibble Length and octet ExtLength 204
Table A.15 – Use of Index formats 205
Table A.16 – Mapping of EventCodes (type 1) 208
Table A.17 – Values of INSTANCE 209
Table A.18 – Values of SOURCE 210
Table A.19 – Values of TYPE 210
Table A.20 – Values of MODE 210
Table B.1 – Direct Parameter page 1 and 2 212
Table B.2 – Types of MasterCommands 213
Table B.3 – Possible values of MasterCycleTime and MinCycleTime 214
Table B.4 – Values of ISDU 214
Table B.5 – Values of SIO 215
Table B.6 – Permitted combinations of BYTE and Length 215
Table B.7 – Implementation rules for parameters and commands 217
Table B.8 – Index assignment of data objects (Device parameter) 218
Table B.9 – Coding of SystemCommand (ISDU) 219
Table B.10 – Data Storage Index assignments 220
Table B.11 – Structure of Index_List 221
Table B.12 – Device locking possibilities 222
Table B.13 – Device status parameter 224
Table B.14 – Detailed Device Status (Index 0x0025) 225
Table B.15 – Time base coding and values of Offset Time 226
Table C.1 – ErrorTypes 228
Table C.2 – Derived ErrorTypes 231
Table D.1 – EventCodes 233
Table D.2 – Basic SDCI EventCodes 235
Table E.1 – BooleanT 236
Table E.2 – BooleanT coding 236
Table E.3 – UIntegerT 237
Table E.4 – IntegerT 237
Table E.5 – IntegerT coding (8 octets) 238
Table E.6 – IntegerT coding (4 octets) 238
Table E.7 – IntegerT coding (2 octets) 238
Table E.8 – IntegerT coding (1 octet) 238
Table E.9 – Float32T 239
Table E.10 – Coding of Float32T 239
Table E.11 – StringT 240
Table E.12 – OctetStringT 240
Table E.13 – TimeT 241
Table E.14 – Coding of TimeT 242
Table E.15 – TimeSpanT 242
Table E.16 – Coding of TimeSpanT 242

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table E.17 – Structuring rules for ArrayT 243
Table E.18 – Example for the access of an ArrayT 243
Table E.19 – Structuring rules for RecordT 244
Table E.20 – Example 1 for the access of a RecordT 244
Table E.21 – Example 2 for the access of a RecordT 244
Table E.22 – Example 3 for the access of a RecordT 245
Table F.1 – Structure of the stored DS data object 247
Table F.2 – Associated header information for stored DS data objects 247
Table G.1 – EMC test conditions for SDCI 248
Table G.2 – EMC test levels 249
Table J.1 – Proper CRC generator polynomials 258

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PROGRAMMABLE CONTROLLERS –

Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

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.

International Standard IEC 61131-9 has been prepared by subcommittee 65B: Measurement and control devices, in collaboration with subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

FDIS	Report on voting
65B/874/FDIS	65B/889/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.

A list of all the parts in the IEC 61131 series, published under the general title *Programmable controllers*, can be found on the IEC website.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

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.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTRODUCTION

0.1 General

IEC 61131-9 is part of a series of standards on programmable controllers and the associated peripherals and should be read in conjunction with the other parts of the series.

Where a conflict exists between this and other IEC standards (except basic safety standards), the provisions of this standard should be considered to govern in the area of programmable controllers and their associated peripherals.

The increased use of micro-controllers embedded in low-cost sensors and actuators has provided opportunities for adding diagnosis and configuration data to support increasing application requirements.

The driving force for the SDCI (IO-Link™[1]) technology is the need of these low-cost sensors and actuators to exchange this diagnosis and configuration data with a controller (PC or PLC) using a low-cost, digital communication technology while maintaining backward compatibility with the current DI/DO signals.

In fieldbus concepts, the SDCI technology defines a generic interface for connecting sensors and actuators to a Master unit, which may be combined with gateway capabilities to become a fieldbus remote I/O node.

Any SDCI compliant Device can be attached to any available interface port of the Master. SDCI compliant Devices perform physical to digital conversion in the Device, and then communicate the result directly in a standard format using "coded switching" of the 24 V I/O signalling line, thus removing the need for different DI, DO, AI, AO modules and a variety of cables.

Physical topology is point-to-point from each Device to the Master using 3 wires over distances up to 20 m. The SDCI physical interface is backward compatible with the usual 24 V I/O signalling specified in IEC 61131-2. Transmission rates of 4,8 kbit/s, 38,4 kbit/s and 230,4 kbit/s are supported.

The Master of the SDCI interface detects, identifies and manages Devices plugged into its ports.

Tools allow the association of Devices with their corresponding electronic I/O Device Descriptions (IODD) and their subsequent configuration to match the application requirements.

The SDCI technology specifies three different levels of diagnostic capabilities: for immediate response by automated needs during the production phase, for medium term response by operator intervention, or for longer term commissioning and maintenance via extended diagnosis information.

The structure of this standard is described in 4.8.

Conformity with IEC 61131-9 cannot be claimed unless the requirements of Annex G are met.

Terms of general use are defined in IEC 61131-1 or in the IEC 60050 series. More specific terms are defined in each part.

---

[1] IO-Link™ is a trade name of the "IO-Link Consortium". This information is given for the convenience of users of this international Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

0.2 Patent declaration

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning the point-to-point serial communication interface for small sensors and actuators as follows, where the [xx] notation indicates the holder of the patent right:

DE 10030845B4 EP 1168271B1 US 6889282B2	[AB]	Fieldbus connecting system for actuators or sensors
EP 1203933 B1	[FE]	Sensor device for measuring at least one variable
DE 10 2004 035 831.1	[SI]	Operational status of a computer system is checked by comparison of actual parameters with reference values and modification to software if needed
DE 102 119 39 A1 US 2003/0200323 A1	[SK]	Coupling apparatus for the coupling of devices to a bus system

IEC takes no position concerning the evidence, validity and scope of these patent rights.

The holders of these patents rights have assured the IEC that they are willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with IEC.

Information may be obtained from:

[AB]	ABB AG Heidelberg Germany
[FE]	Festo AG Esslingen Germany
[SI]	Siemens AG Otto-Hahn-Ring 6 81739 Munich Germany
[SK]	Sick AG Waldkirch Germany

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

PROGRAMMABLE CONTROLLERS –

Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)

1 Scope

This part of IEC 61131 specifies a single-drop digital communication interface technology for small sensors and actuators SDCI (commonly known as IO-Link™[2]), which extends the traditional digital input and digital output interfaces as defined in IEC 61131-2 towards a point-to-point communication link. This technology enables the transfer of parameters to Devices and the delivery of diagnostic information from the Devices to the automation system.

This technology is mainly intended for use with simple sensors and actuators in factory automation, which include small and cost-effective microcontrollers.

This part specifies the SDCI communication services and protocol (physical layer, data link layer and application layer in accordance with the ISO/OSI reference model) for both SDCI Masters and Devices.

This part also includes EMC test requirements.

This part does not cover communication interfaces or systems incorporating multiple point or multiple drop linkages, or integration of SDCI into higher level systems such as fieldbuses.

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 60947-5-2, *Low-voltage switchgear and controlgear – Part 5-2: Control circuit devices and switching elements – Proximity switches*

IEC 61000-4-2, *Electromagnetic compatibility (EMC) – Part 4-2: Testing and measurement techniques – Electrostatic discharge immunity test*

IEC 61000-4-3, *Electromagnetic compatibility (EMC) – Part 4-3: Testing and measurement techniques – Radiated, radio-frequency, electromagnetic field immunity test*

IEC 61000-4-4, *Electromagnetic compatibility (EMC) – Part 4-4: Testing and measurement techniques – Electrical fast transient/burst immunity test*

IEC 61000-4-5, *Electromagnetic compatibility (EMC) – Part 4-5: Testing and measurement techniques – Surge immunity test*

---

[2] IO-Link™ is a trade name of the "IO-Link Consortium". This information is given for the convenience of users of this international Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

IEC 61000-4-6, *Electromagnetic compatibility (EMC) – Part 4-6: Testing and measurement techniques – Immunity to conducted disturbances, induced by radio-frequency fields*

IEC 61000-4-11, *Electromagnetic compatibility (EMC) – Part 4-11: Testing and measurement techniques – Voltage dips, short interruptions and voltage variations immunity tests*

IEC 61000-6-2, *Electromagnetic compatibility (EMC) – Part 6-2: Generic standards – Immunity for industrial environments*

IEC 61000-6-4, *Electromagnetic compatibility (EMC) – Part 6-4: Generic standards – Emission standard for industrial environments*

IEC 61076-2-101, *Connectors for electronic equipment – Product requirements – Part 2-101: Circular connectors – Detail specification for M12 connectors with screw-locking*

IEC 61131-1, *Programmable controllers – Part 1: General information*

IEC 61131-2, *Programmable controllers – Part 2: Equipment requirements and tests*

IEC/TR 62390, *Common automation device – Profile guideline*

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 2022, *Information technology – Character code structure and extension techniques*

ISO/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO/IEC 19505 (all parts), *Information technology – Object Management Group Unified Modeling Language (OMG UML)*

ISO 1177, *Information processing – Character structure for start/stop and synchronous character oriented transmission*

IEEE Std 754-2008, *IEEE Standard for Floating-Point Arithmetic*

Internet Engineering Task Force (IETF): RFC 5905 – *Network Time Protocol Version 4: Protocol and Algorithms Specification;* available at < www.ietf.org >

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61131-1 and IEC 61131-2, as well as the following apply.

3.1.1
address
part of the M-sequence control to reference data within data categories of a communication channel

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.2
application layer
AL
<SDCI> part of the protocol responsible for the transmission of Process Data objects and On-Request Data objects

3.1.3
block parameter
consistent parameter access via multiple Indices or Subindices

3.1.4
checksum
<SDCI> complementary part of the overall data integrity measures in the data link layer in addition to the UART parity bit

3.1.5
CHKPDU
integrity protection data within an ISDU communication channel generated through XOR processing the octets of a request or response

3.1.6
coded switching
SDCI communication, based on the standard binary signal levels of IEC 61131-2

3.1.7
COM1
SDCI communication mode with transmission rate of 4,8 kbit/s

3.1.8
COM2
SDCI communication mode with transmission rate of 38,4 kbit/s

3.1.9
COM3
SDCI communication mode with transmission rate of 230,4 kbit/s

3.1.10
COMx
one out of three possible SDCI communication modes COM1, COM2, or COM3

3.1.11
communication channel
logical connection between Master and Device

Note 1 to entry: Four communication channels are defined: process channel, page and ISDU channel (for parameters), and diagnosis channel.

3.1.12
communication error
unexpected disturbance of the SDCI transmission protocol

3.1.13
cycle time
time to transmit an M-sequence between a Master and its Device including the following idle time

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.14
Device
single passive peer to a Master such as a sensor or actuator

Note 1 to entry: Uppercase "Device" is used for SDCI equipment, while lowercase "device" is used in a generic manner.

3.1.15
Direct Parameters
directly (page) addressed parameters transferred acyclically via the page communication channel without acknowledgement

3.1.16
dynamic parameter
part of a Device's parameter set defined by on-board user interfaces such as teach-in buttons or control panels in addition to the static parameters

3.1.17
Event
instance of a change of conditions in a Device

Note 1 to entry: Uppercase "Event" is used for SDCI Events, while lowercase "event" is used in a generic manner.

Note 2 to entry: An Event is indicated via the Event flag within the Device's status cyclic information, then acyclic transfer of Event data (typically diagnosis information) is conveyed through the diagnosis communication channel.

3.1.18
fallback
transition of a port from coded switching to switching signal mode

3.1.19
inspection level
degree of verification for the Device identity

3.1.20
interleave
segmented cyclic data exchange for Process Data with more than 2 octets through subsequent cycles

3.1.21
ISDU
indexed service data unit used for acyclic acknowledged transmission of parameters that can be segmented in a number of M-sequences

3.1.22
legacy Device or Master
Device or Master designed in accordance with [8][3]

3.1.23
M-sequence
sequence of two messages comprising a Master message and its subsequent Device message

3.1.24
M-sequence control
first octet in a Master message indicating the read/write operation, the type of the communication channel, and the address, for example offset or flow control

---

3 Numbers in square brackets refer to the Bibliography.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.25
M-sequence error
unexpected or wrong message content, or no response

3.1.26
M-sequence type
one particular M-sequence format out of a set of specified M-sequence formats

3.1.27
Master
active peer connected through ports to one up to *n* Devices and which provides an interface to the gateway to the upper level communication systems or PLCs

Note 1 to entry: Uppercase "Master" is used for SDCI equipment, while lowercase "master" is used in a generic manner.

3.1.28
message
<SDCI> sequence of UART frames transferred either from a Master to its Device or vice versa following the rules of the SDCI protocol

3.1.29
On-request Data
acyclically transmitted data upon request of the Master application consisting of parameters or Event data

3.1.30
physical layer
first layer of the ISO-OSI reference model, which provides the mechanical, electrical, functional and procedural means to activate, maintain, and de-activate physical connections for bit transmission between data-link entities

Note 1 to entry: Physical layer also provides means for wake-up and fallback procedures.

[SOURCE: ISO/IEC 7498-1:1994, 7.7.2, modified – text extracted from subclause, note added]

3.1.31
port
communication medium interface of the Master to one Device

3.1.32
port operating mode
state of a Master's port that can be either INACTIVE, DO, DI, FIXEDMODE, or SCANMODE

3.1.33
Process Data
input or output values from or to a discrete or continuous automation process cyclically transferred with high priority and in a configured schedule automatically after start-up of a Master

3.1.34
Process Data cycle
complete transfer of all Process Data from or to an individual Device that may comprise several cycles in case of segmentation (interleave)

3.1.35
single parameter
independent parameter access via one single Index or Subindex

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.36
SIO
port operation mode in accordance with digital input and output defined in IEC 61131-2 that is established after power-up or fallback or unsuccessful communication attempts

3.1.37
static parameter
part of a Device's parameter set to be saved in a Master for the case of replacement without engineering tools

3.1.38
switching signal
binary signal from or to a Device when in SIO mode (as opposed to the "coded switching" SDCI communication)

3.1.39
system management
SM
<SDCI> means to control and coordinate the internal communication layers and the exceptions within the Master and its ports, and within each Device

3.1.40
UART frame
<SDCI> bit sequence starting with a start bit, followed by eight bits carrying a data octet, followed by an even parity bit and ending with one stop bit

3.1.41
wake-up
procedure for causing a Device to change its mode from SIO to SDCI

3.1.42
wake-up request
WURQ
physical layer service used by the Master to initiate wake-up of a Device, and put it in a receive ready state

3.2 Symbols and abbreviated terms

Δf_{DTRM}	permissible deviation from data transfer rate (measured in %)
ΔVS	power supply ripple (measured in V)
AL	application layer
BEP	bit error probability
C/Q	connection for communication (C) or switching (Q) signal (SIO)
CL_{eff}	effective total cable capacity (measured in nF)
CQ	input capacity at C/Q connection (measured in nF)
DI	digital input
DL	data link layer
DO	digital output
f_{DTR}	data transfer rate (measured in bit/s)
H/L	high/low signal at receiver output
I/O	input / output
ILL	input load current at input C/Q to $V0$ (measured in A)
IODD	IO Device Description (see 10.8)
IQ	driver current in saturated operating status ON (measured in A)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

IQH	driver current on high-side driver in saturated operating status ON (measured in A)
IQL	driver current on low-side driver in saturated operating status ON (measured in A)
$IQPK$	maximum driver current in unsaturated operating status ON (measured in A)
$IQPKH$	maximum driver current on high-side driver in unsaturated operating status ON (measured in A)
$IQPKL$	maximum driver current on low-side driver in unsaturated operating status ON (measured in A)
IQQ	quiescent current at input C/Q to $V0$ with inactive output drivers (measured in A)
IQ_{WU}	amplitude of Master's wake-up request current (measured in A)
IS	supply current at $V+$ (measured in A)
$ISIR$	current pulse supply capability at $V+$ (measured in A)
LED	light emitting diode
L-	power supply (-)
L+	power supply (+)
N24	24 V extra power supply (-)
n_{WU}	wake-up retry count
On/Off	driver's ON/OFF switching signal
OD	on-request data
OVD	signal overload detect
P24	24 V extra power supply (+)
PD	process data
PDCT	port and device configuration tool
PL	physical layer
PLC	programmable logic controller
PS	power supply (measured in V)
r	time to reach a stable level with reference to the beginning of the start bit (measured in T_{BIT})
RL_{eff}	loop resistance of cable (measured in Ω)
s	time to exit a stable level with reference to the beginning of the start bit (measured in T_{BIT})
SDCI	single-drop digital communication interface
SIO	standard input output (digital switching mode) [IEC 61131-2]
SM	system management
t_1	UART frame transfer delay on Master (measured in T_{BIT})
t_2	UART frame transfer delay on Device (measured in T_{BIT})
t_A	response delay on Device (measured in T_{BIT})
T_{BIT}	bit time (measured in s)
t_{CYC}	cycle time on M-sequence level (measured in s)
t_{DF}	fall time (measured in s)
T_{DMT}	delay time while establishing Master port communication (measured in T_{BIT})
T_{DR}	rise time (measured in s)
T_{DSIO}	delay time on device for transition to SIO mode following wake-up request (measured in s)
T_{DWU}	wake-up retry delay (measured in s)
$t_{M\text{-}sequence}$	M-sequence duration (measured in T_{BIT})
t_{idle}	idle time between two M-sequences (measured in s)
t_H	detection time for high level (measured in s)
t_L	detection time for low level (measured in s)
t_{ND}	noise suppression time (measured in s)
T_{RDL}	wake-up readiness following power ON (measured in s)
T_{REN}	receive enable (measured in s)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

T_{SD}	device detect time (measured in s)
T_{WU}	pulse duration of wake-up request (measured in s)
UART	universal asynchronous receiver transmitter
UML	Unified Modelling Language [ISO/IEC 19505]
$V+$	voltage at L+
$V0$	voltage at L-
$VD+_L$	voltage drop on the line between the L+ connections on Master and Device (measured in V)
$VD0_L$	voltage drop on the line between the L- connections on Master and Device (measured in V)
VDQ_L	voltage drop on the line between the C/Q connections on Master and Device (measured in V)
$VHYS$	hysteresis of receiver threshold voltage (measured in V)
VI	input voltage at connection C/Q with reference to $V0$ (measured in V)
VIH	input voltage range at connection C/Q for high signal (measured in V)
VIL	input voltage range at connection C/Q for low signal (measured in V)
VRQ	residual voltage on driver in saturated operating status ON (measured in V)
$VRQH$	residual voltage on high-side driver in operating status ON (measured in V)
$VRQL$	residual voltage on low-side driver in saturated operating status ON (measured in V)
VTH	threshold voltage of receiver with reference to $V0$ (measured in V)
$VTHH$	threshold voltage of receiver for safe detection of a high signal (measured in V)
$VTHL$	threshold voltage of receiver for safe detection of a low signal (measured in V)
WURQ	wake-up request pulse

3.3 Conventions

3.3.1 General

The service model, service primitives, and the diagrams shown in this standard are entirely abstract descriptions. The implementation of the services may reflect individual issues and can be different.

3.3.2 Service parameters

Service primitives are used to represent service provider/consumer interactions (ISO/IEC 10731). They convey parameters which indicate the information available in the provider/ consumer interaction. In any particular interface, not each and every parameter needs to be explicitly stated.

The service specification in this standard uses a tabular format to describe the component parameters of the service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns:

1) parameter name;
2) request primitive (.req);
3) indication primitive (.ind);
4) response primitive (.rsp); and
5) confirmation primitive (.cnf).

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column.

M Parameter is mandatory for the primitive.

U Parameter is a user option and can or cannot be provided depending on dynamic usage of the service user. When not provided a default value for the parameter is assumed.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C Parameter is conditional upon other parameters or upon the environment of the service user.

– Parameter is never present.

S Parameter is a selected item.

Some entries are further qualified by items in brackets. These may be:

a) a parameter-specific constraint "(=)" indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table;

b) an indication that some note applies to the entry "(n)" indicates that the following note "n" contains additional information related to the parameter and its use.

3.3.3 Service procedures

The procedures are defined in terms of:

- the interactions between application entities through the exchange of protocol data units; and
- the interactions between a communication layer service provider and a communication layer service consumer in the same system through the invocation of service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the communication layers.

3.3.4 Service attributes

The nature of the different (Master and Device) services is characterized by attributes. All services are defined from the view of the affected layer towards the layer above.

I Initiator of a service (towards the layer above)

R Receiver (responder) of a service (from the layer above)

3.3.5 Figures

For figures that show the structure and services of protocol layers, the following conventions are used:

- an arrow with just a service name represents both a request and the corresponding confirmation, with the request being in the direction of the arrow;
- a request without confirmation, as well as all indications and responses are labelled as such (i.e. service.req, service.ind, service.rsp).

Figure 1 shows the example of a confirmed service.

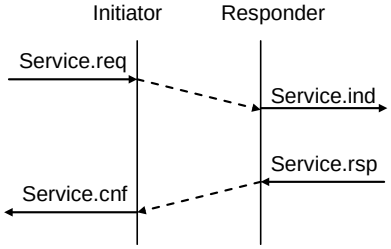
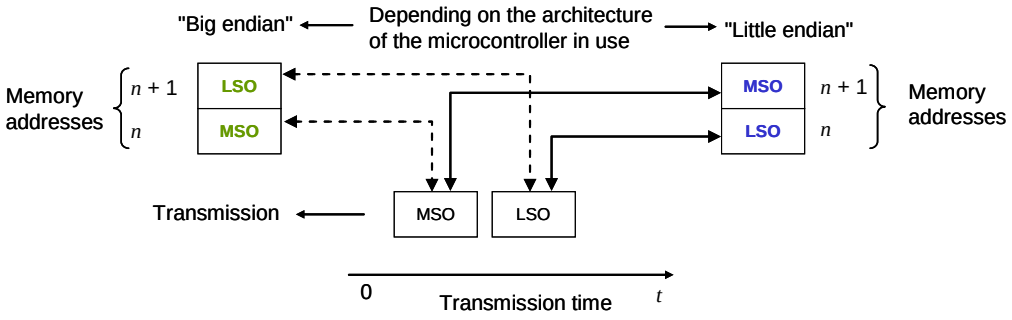


Figure 1 – Example of a confirmed service

3.3.6 Transmission octet order

Figure 2 shows how WORD based data types are transferred from memory to transmission medium and vice versa (i.e. most significant octet transmitted first, see 7.3.3.2 and 7.3.6.1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Key

MSO = Most significant octet
LSO = Least significant octet

Figure 2 – Memory storage and transmission order for WORD based data types

3.3.7 Behavioral descriptions

For the behavioral descriptions, the notations of UML 2 (ISO/IEC 19505) are used (e.g. state, sequence, activity, timing diagrams, guard conditions). The layout of the associated state-transition tables is following IEC/TR 62390.

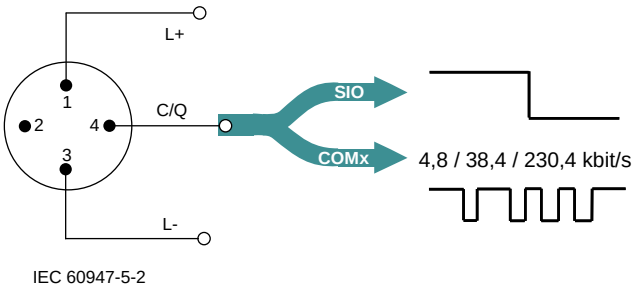
Due to design tool restrictions the following exceptions apply. For state diagrams, a service parameter (in capital letters) is attached to the service name via an underscore character, such as for example in DL_SetMode_INACTIVE. For sequence diagrams, the service primitive is attached via an underscore character instead of a dot, and the service parameter is added in parenthesis, such as for example in DL_Event_ind (OPERATE). Timing constraints are labelled "tm(time in ms)".

Asynchronously received service calls are not modelled in detail within state diagrams.

4 Overview of SDCI (IO-Link™[4])

4.1 Purpose of technology

Figure 3 shows the basic concept of SDCI.



Pin	Signal	Definition	Standard
1	L+	24 V	IEC 61131-2
2	I/Q	Not connected, DI, or DO	IEC 61131-2
3	L-	0 V	IEC 61131-2
4	Q	"Switching signal" DI, DO (SIO)	IEC 61131-2
	C	"Coded switching" (COM1, COM2, COM3)	IEC 61131-9

Figure 3 – SDCI compatibility with IEC 61131-2

[4] IO-Link™ is a trade name of the "IO-Link Consortium". This information is given for the convenience of users of this international Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The single-drop digital communication interface technology for small sensors and actuators SDCI (commonly known as IO-Link™) defines a migration path from the existing digital input and digital output interfaces for switching 24 V Devices as defined in IEC 61131-2 towards a point-to-point communication link. Thus, for example, digital I/O modules in existing fieldbus peripherals can be replaced by SDCI Master modules providing both classic DI/DO interfaces and SDCI. Analog transmission technology can be replaced by SDCI combining its robustness, parameterization, and diagnostic features with the saving of digital/analog and analog/digital conversion efforts.

4.2 Positioning within the automation hierarchy

Figure 4 shows the domain of the SDCI technology within the automation hierarchy.

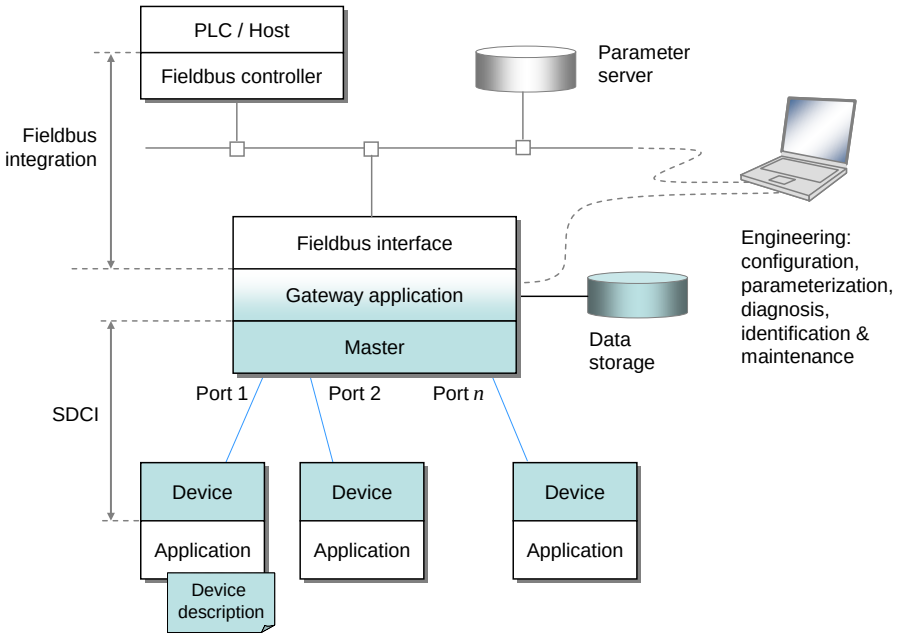


Figure 4 – Domain of the SDCI technology within the automation hierarchy

The SDCI technology defines a generic interface for connecting sensors and actuators to a Master unit, which may be combined with gateway capabilities to become a fieldbus remote I/O node.

Starting point for the design of SDCI is the classic 24 V digital input (DI) defined in IEC 61131-2 and output interface (DO) specified in Table 6. Thus, SDCI offers connectivity of classic 24 V sensors ("switching signals") as a default operational mode. Additional connectivity is provided for actuators when a port has been configured into "single-drop communication mode".

Many sensors and actuators nowadays are already equipped with microcontrollers offering a UART interface that can be extended by addition of a few hardware components and protocol software to support SDCI communication. This second operational mode uses "coded switching" of the 24 V I/O signalling line. Once activated, the SDCI mode supports parameterization, cyclic data exchange, diagnosis reporting, identification & maintenance information, and external parameter storage for Device backup and fast reload of replacement devices. Sensors and actuators with SDCI capability are referred to as "Devices" in this standard. To improve start-up performance these Devices usually provide non-volatile storage for parameters.

NOTE Configuration and parameterization of Devices is supported through an XML-based device description (see [6]), which is not part of this standard.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

4.3 Wiring, connectors and power

The default connection (port class A) comprises 4 pins (see Figure 3). The default wiring for port class A complies with IEC 60947-5-2 and uses only three wires for 24 V, 0 V, and a signal line. The fourth wire may be used as an additional signal line complying with IEC 61131-2.

Five pins connections (port class B) are specified for Devices requiring additional power from an independant 24 V power supply.

NOTE A port class A Device using the fourth wire is not compatible with a port class B Master.

Maximum length of cables is 20 m, shielding is not required.

4.4 Communication features of SDCI

The generic Device model is shown in Figure 5 and explained in the following paragraphs.

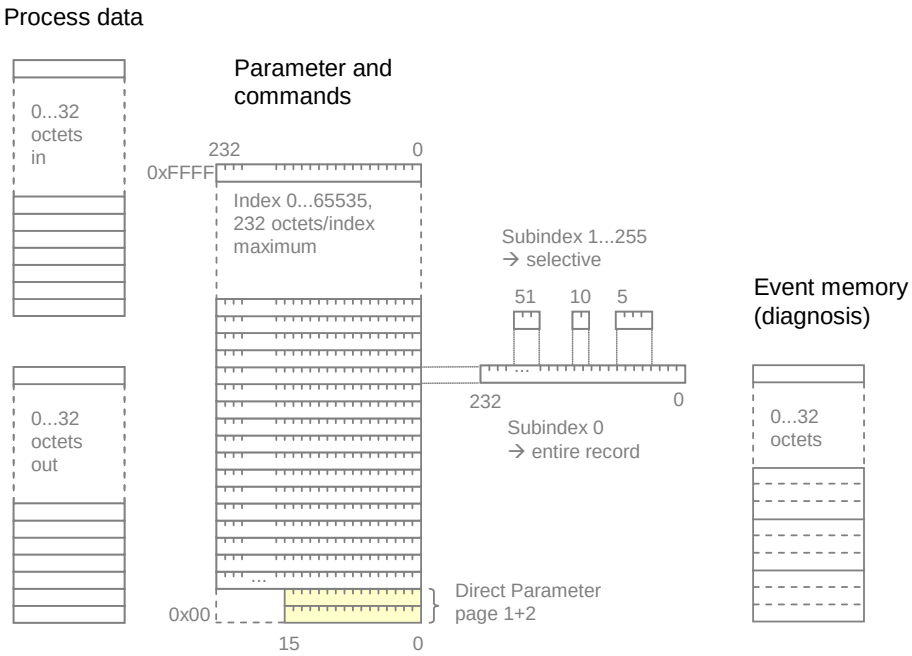


Figure 5 – Generic Device model for SDCI (Master's view)

A Device may receive Process Data (out) to control a discrete or continuous automation process or send Process Data (in) representing its current state or measurement values. The Device usually provides parameters enabling the user to configure its functions to satisfy particular needs. To support this case a large parameter space is defined with access via an Index (0 to 65 535; with a predefined organization) and a Subindex (0 to 255).

The first two index entries 0 and 1 are reserved for the Direct Parameter page 1 and 2 with a maximum of 16 octets each. Parameter page 1 is mainly dedicated to Master commands such as Device startup and fallback, retrieval of Device specific operational and identification information. Parameter page 2 allows for a maximum of 16 octets of Device specific parameters.

The other indices (2 to 65 535) each allow access to one record having a maximum size of 232 octets. Subindex 0 specifies transmission of the complete record addressed by the Index, other subindices specify transfer of selected data items within the record.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Within a record, individual data items may start on any bit offset, and their length may range from 1 bit to 232 octets, but the total number of data items in the record cannot exceed 255. The organization of data items within a record is specified in the IO Device Description (IODD).

All changes of Device condition that require reporting or intervention are stored within an Event memory before transmission. An Event flag is then set in the cyclic data exchange to indicate the existence of an Event.

Communication between a Master and a Device is point-to-point and is based on the principle of a Master first sending a request message and then a Device sending a response message (see Figure 36). Both messages together are called an M-sequence. Several M-sequence types are defined to support user requirements for data transmission (see Figure 37).

Data of various categories are transmitted through separate communication channels within the data link layer, as shown in Figure 6.

- Operational data such as Device inputs and outputs is transmitted through a process channel using cyclic transfer. Operational data may also be associated with qualifiers such as valid/invalid.
- Configuration and maintenance parameters are transmitted using acyclic transfers. A page channel is provided for direct access to parameter pages 1 and 2, and an ISDU channel is used for accessing additional parameters and commands.
- Device events are transmitted using acyclic transfers through a diagnostic channel. Device events are reported using 3 severity levels, error, warning, and notification.

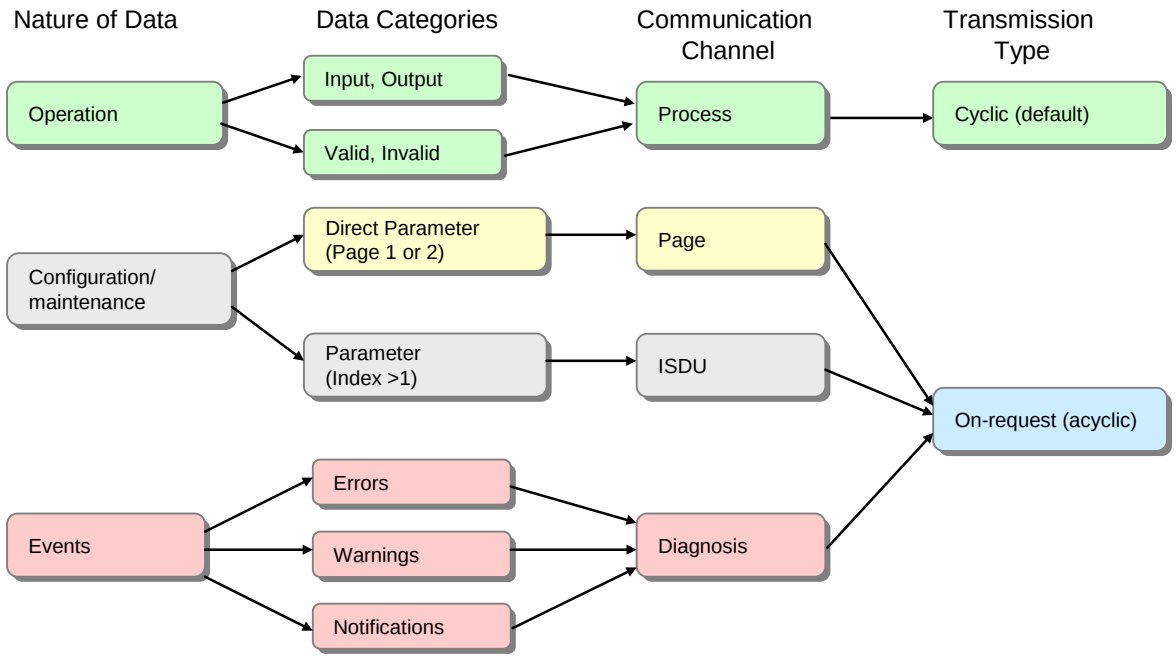


Figure 6 – Relationship between nature of data and transmission types

The first octet of a Master message controls the data transfer direction (read/write) and the type of communication channel.

Figure 7 shows each port of a Master has its own data link layer which interfaces to a common master application layer. Within the application layer, the services of the data link layer are translated into actions on Process Data objects (input/output), On-request Data objects (read/write), and events. Master applications include a Configuration Manager (CM),

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Data Storage mechanism (DS), Diagnosis Unit (DU), On-request Data Exchange (ODE), and a Process Data Exchange (PDE).

System management checks identification of the connected Devices and adjusts ports and Devices to match the chosen configuration and the properties of the connected Devices. It controls the state machines in the application (AL) and data link layers (DL), for example at start-up.

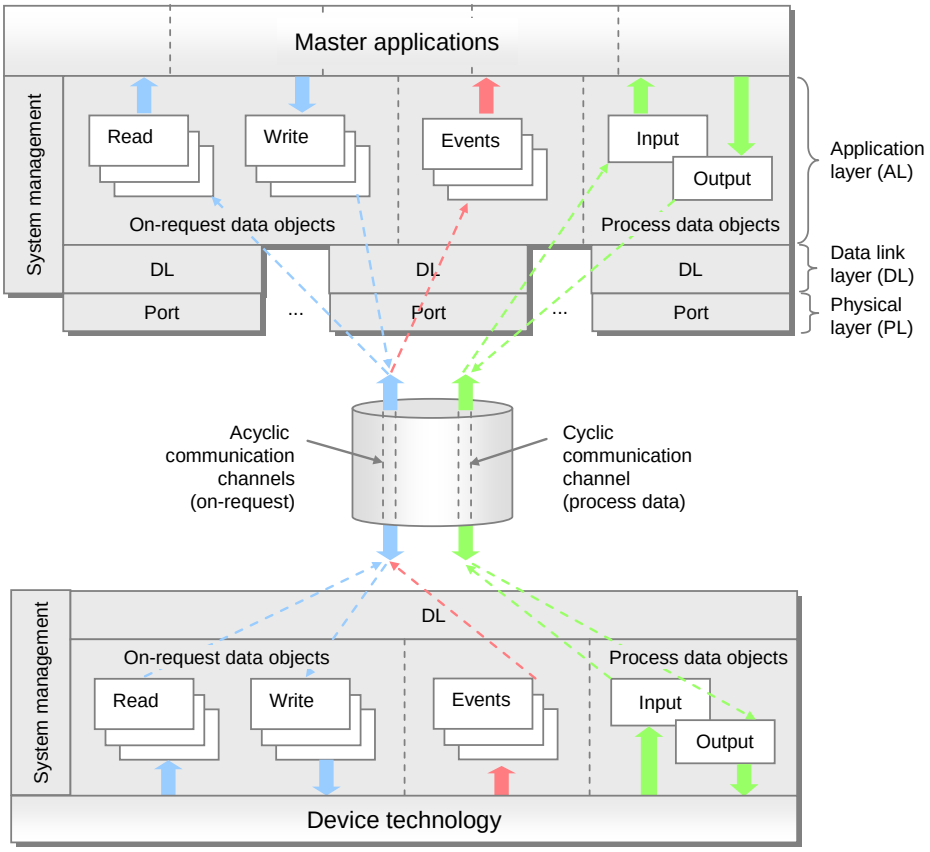


Figure 7 – Object transfer at the application layer level (AL)

4.5 Role of a Master

A Master accommodates 1 to *n* ports and their associated data link layers. During start-up it changes the ports to the user-selected port modes, which can be INACTIVE, DI, DO, FIXEDMODE, or SCANMODE. If communication is requested, the Master uses a special wake-up current pulse to initiate communication with the Device. The Master then auto-adjusts the transmission rate to COM1, COM2, or COM3 (see Table 8) and checks the "personality" of the connected Device, i.e. its VendorID, DeviceID, and communication properties.

If there is a mismatch between the Device parameters and the stored parameter set within the Master, the parameters in the Device are overwritten (see 11.3) or the stored parameters within the master are updated depending on configuration.

It is also possible to start a device in DI mode, switch to SDCI communication for configuration and parameterization and then use the fallback command (see 11.8.5) to switch back to DI mode for normal operation.

Coordination of the ports is also a task of the Master which the user can configure through the selection of port cycle modes. In "FreeRunning" mode, each port defines its own cycle based on the properties of the connected Device. In "MessageSync" mode, messages sent on the

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

connected ports start at the same time or in a defined staggered manner. In "FixedValue" mode, each port uses a user-defined fixed cycle time (see 11.2.2.2).

The Master is responsible for the assembling and disassembling of all data from or to the Devices (see Clause 11).

The Master provides a Data Storage area of at least 2 048 octets per Device for backup of Device data (see 11.3). The Master may combine this Device data together with all other relevant data for its own operation, and make this data available for higher level applications for Master backup purpose or recipe control (see 11.8.3).

4.6 SDCI configuration

Engineering support for a Master is usually provided by a Port and Device Configuration Tool (PDCT). The PDCT configures both port properties and Device properties (see parameters shown in Figure 5). It combines both an interpreter of the I/O Device Description (IODD) and a configurator (see 11.7). The IODD provides all the necessary properties to establish communication and the necessary parameters and their boundaries to establish the desired function of a sensor or actuator. The PDCT also supports the compilation of the Process Data for propagation on the fieldbus and vice versa.

4.7 Mapping to fieldbuses

Integration of a Master within a fieldbus system, i.e. the definition of gateway fuctions for exchanging data with higher level entities on a fieldbus, is out of the scope of this standard.

EXAMPLE These functions include mapping of the Process Data exchange, realization of program-controlled parameterization or a remote parameter server, or the propagation of diagnosis information.

The integration of a PDCT into engineering tools of a particular fieldbus is out of the scope of this standard.

4.8 Standard structure

Figure 8 shows the logical structure of the Master and Device. Clause 5 specifies the Physical Layer (PL) of SDCI, Clause 6 specifies details of the SIO mode. Clause 7 specifies Data Link Layer (DL) services, protocol, wake-up, M-sequences, and the DL layer handlers. Clause 8 specifies the services and the protocol of the Application Layer (AL) and Clause 9 the System Management responsibilities (SM).

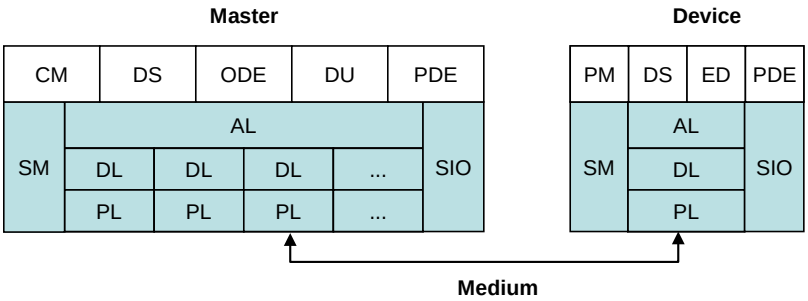


Figure 8 – Logical structure of Master and Device

Clause 10 specifies Device applications and features. These include Process Data Exchange (PDE), Parameter Management (PM), Data Storage (DS), and Event Dispatcher (ED). Technology specific applications are not part of this standard. They may be specified in profiles for particular Device families.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Clause 11 specifies Master applications and features. These include Process Data Exchange (PDE), On-request Data Exchange (ODE), Configuration Management (CM), Data Storage (DS) and Diagnosis Unit (DU).

Several normative and informative annexes are included. Annex A defines the available M-sequence types. Annex B describes the parameters of the Direct Parameter page and the fixed Device parameters. Annex C lists the error types in case of acyclic transmissions and Annex D the EventCodes (diagnosis information of Devices). Annex E specifies the available basic and composite data types. Annex F defines the structure of Data Storage objects. Annex G deals with conformity and electromagnetic compatibility test requirements and Annex H provides graphs of residual error probabilities, demonstrating the level of SDCI's data integrity. The informative Annex I provides an example of the sequence of acyclic data transmissions. The informative Annex J explains two recommended methods for detecting parameter changes in the context of Data Storage.

5 Physical Layer (PL)

5.1 General

5.1.1 Basics

The 3-wire connection system of SDCI is based on the specifications in IEC 60947-5-2. The three lines are used as follows: (L+) for the 24 V power supply, (L-) for the ground line, and (C/Q) for the switching signal (Q) or SDCI communication (C), as shown in Figure 9.

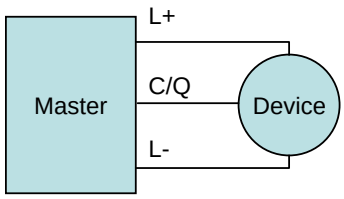


Figure 9 – Three wire connection system

NOTE 1 Binary sensors compliant with IEC 60947-5-2 are compatible with the SDCI 3-wire connection system (including from a power consumption point of view).

Support of the SDCI 3-wire connection system is mandatory for Master. Ports with this characteristic are called port class A.

Port class A uses a four pin connector. The fourth wire may be used as an additional signal line complying with IEC 61131-2. Its support is optional in both Masters and Devices.

Five wire connections (port class B) are specified for Devices requiring additional power from an independant 24 V power supply (see 5.5.1).

NOTE 2 A port class A Device using the fourth wire is not compatible with a port class B Master.

5.1.2 Topology

The SDCI system topology uses point-to-point links between a Master and its Devices as shown in Figure 10. The Master may have multiple ports for the connection of Devices. Only one Device shall be connected to each port.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

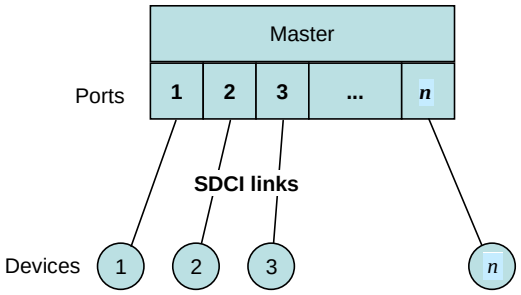


Figure 10 – Topology of SDCI

5.2 Physical layer services

5.2.1 Overview

Figure 11 shows an overview of the Master's physical layer and its service primitives.

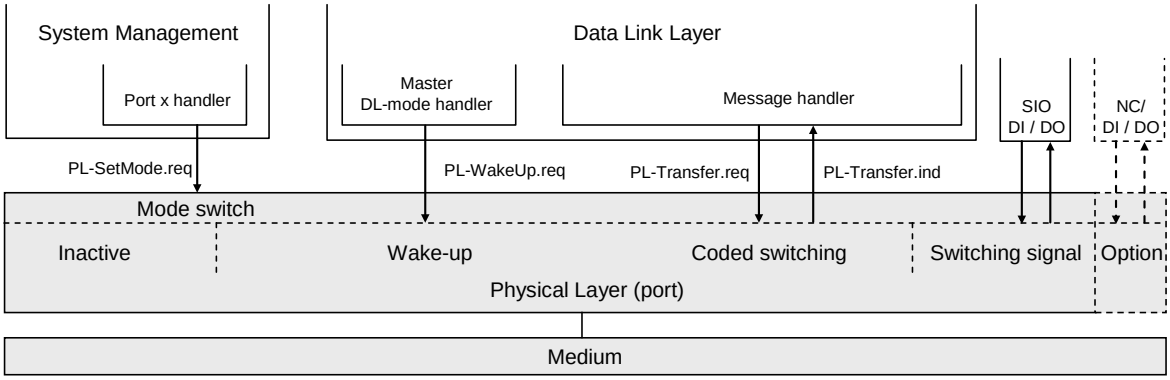


Figure 11 – Physical layer (Master)

The physical layer specifies the operation of the C/Q line in Figure 3 and the associated line driver (transmitter) and receiver of a particular port. The Master operates this line in three main modes (see Figure 11): inactive, "Switching signal" (DI/DO), or "Coded switching" (COMx). The service PL-SetMode.req is responsible for switching into one of these modes.

If the port is in inactive mode, the C/Q line shall be high impedance (floating). In SIO mode, the port can be used as a standard input or output interface according to the definitions of IEC 61131-2 or in Table 6 respectively. The communication layers of SDCI are bypassed as shown in Figure 11; the signals are directly processed within the Master application. In SDCI mode, the service PL_WakeUp.req creates a special signal pattern (current pulse) that can be detected by an SDCI enabled Device connected to this port (see 5.3.3.3).

Figure 12 shows an overview of the Device's physical layer and its service primitives.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

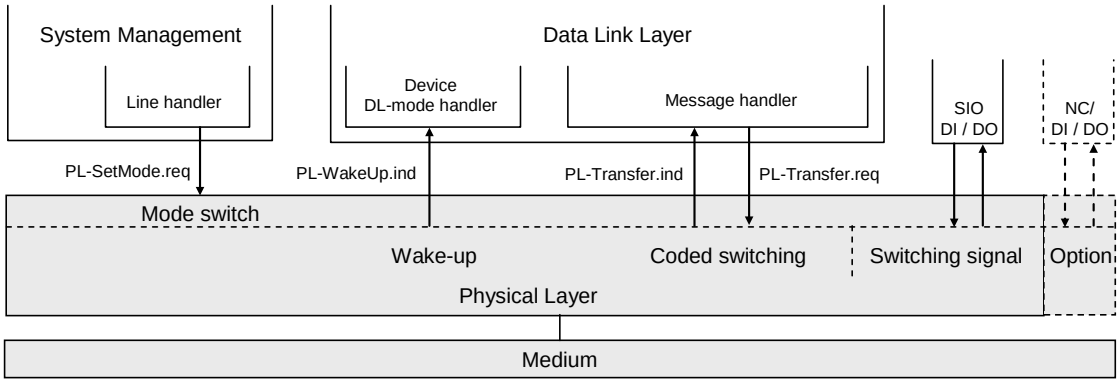


Figure 12 – Physical layer (Device)

The physical layer of a Device according to Figure 12 follows the same principle, except that there is no inactive state. By default at power on or cable reconnection, the Device shall operate in the SIO mode, as a digital input. The Device shall always be able to detect a wake-up current pulse (wake-up request). The service PL_WakeUp.ind reports successful detection of the wake-up request (usually a microcontroller interrupt), which is required for the Device to switch to the SDCI mode.

A special MasterCommand (fallback) sent via SDCI causes the Device to switch back to SIO mode.

Subsequently, the services are specified that are provided by the PL to System Management and to the Data Link Layer (see Figure 83 and Figure 94 for a complete overview of all the services). Table 1 lists the assignments of Master and Device to their roles as initiator or receiver for the individual PL services.

Table 1 – Service assignments of Master and Device

Service name	Master	Device
PL-SetMode	R	R
PL-WakeUp	R	I
PL-Transfer	I / R	R / I
Key (see 3.3.4) I Initiator of service R Receiver (Responder) of service		

5.2.2 PL services

5.2.2.1 PL_SetMode

The PL-SetMode service is used to setup the electrical characteristics and configurations of the Physical Layer. The parameters of the service primitives are listed in Table 2.

Table 2 – PL_SetMode

Parameter name	.req
Argument	M
TargetMode	M

Argument
The service-specific parameters of the service request are transmitted in the argument.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TargetMode
This parameter indicates the requested operation mode

Permitted values:
INACTIVE (C/Q line in high impedance),
DI (C/Q line in digital input mode),
DO (C/Q line in digital output mode),
COM1 (C/Q line in COM1 mode),
COM2 (C/Q line in COM2 mode),
COM3 (C/Q line in COM3 mode)

5.2.2.2 PL_WakeUp

The PL-WakeUp service initiates or indicates a specific sequence which prepares the Physical Layer to send and receive communication requests (see 5.3.3.3). This unconfirmed service has no parameters. Its success can only be verified by a Master by attempting to communicate with the Device. The service primitives are listed in Table 3.

Table 3 – PL_WakeUp

Parameter name	.req	.ind
<none>		

5.2.2.3 PL_Transfer

The PL-Transfer service is used to exchange the SDCI data between Data Link Layer and Physical Layer. The parameters of the service primitives are listed in Table 4.

Table 4 – PL_Transfer

Parameter name	.req	ind.
Argument		
Data	M	M
Result (+)		S
Result (-)		S
Status		M

Argument
The service-specific parameters of the service request are transmitted in the argument.

Data
This parameter contains the data value which is transferred over the SDCI interface.

Permitted values: 0…255

Result (+):
This selection parameter indicates that the service request has been executed successfully.

Result (-):
This selection parameter indicates that the service request failed.

Status
This parameter contains supplementary information on the transfer status.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permitted values:
PARITY_ERROR (UART detected a parity error),
FRAMING_ERROR (invalid UART stop bit detected),
OVERRUN (octet collision within the UART)

5.3 Transmitter/Receiver

5.3.1 Description method

The physical layer is specified by means of electrical and timing requirements. Electrical requirements specify signal levels and currents separately for Master and Device in the form of reference schematics. Timing requirements specify the signal transmission process (specifically the receiver) and a special signal detection function.

5.3.2 Electrical requirements

5.3.2.1 General

The line driver is specified by a reference schematic corresponding to Figure 13. On the Master side, a transmitter comprises a combination of two line drivers and one current sink. On the Device side, in its simplest form, the transmitter takes the form of a p-switching driver. As an option there can be an additional n-switching or non-switching driver (this also allows the option of push-pull output operation).

In operating status ON the descriptive variables are the residual voltage *VRQ*, the standard driver current *IQ*, and the peak current *IQPK*. The source is controlled by the On/Off signal. An overload current event is indicated at the “Overload” output (OVD). This feature can be used for the current pulse detection (wake-up).

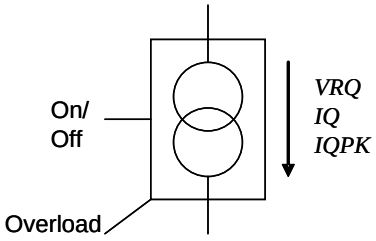


Figure 13 – Line driver reference schematics

The receiver is specified by a reference schematic according to Figure 14. It performs the function of a comparator and is specified by its switching thresholds *VTH* and a hysteresis *VHYS* between the switching thresholds. The output indicates the logic level (High or Low) at the receiver input.

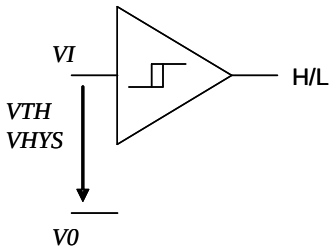
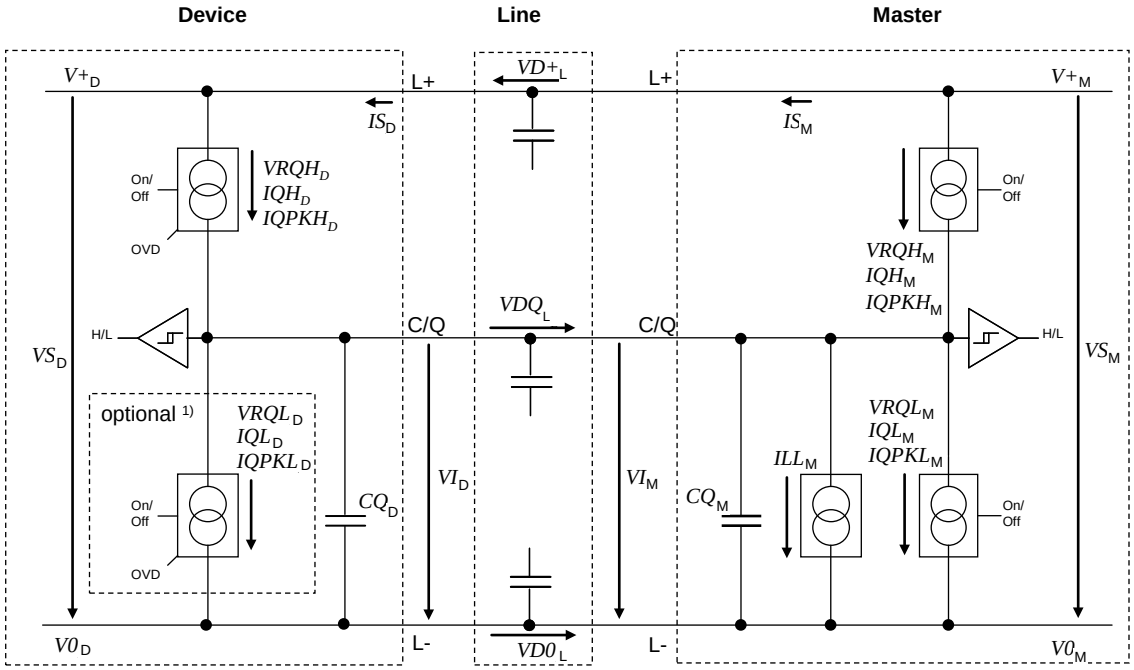


Figure 14 – Receiver reference schematics

Figure 15 shows the reference schematics for the interconnection of Master and Device for the SDCI 3-wire connection system.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



1) Optional:low-side driver (push-pull only)

Figure 15 – Reference schematics for SDCI 3-wire connection system

The subsequent illustrations and parameter tables refer to the voltage level definitions in Figure 16. The parameter indices refer to the Master (M), Device (D) or line (L). The voltage drops on the line $VD+_L$, VDQ_L and $VD0_L$ are implicitely specified in 5.5 through cable parameters.

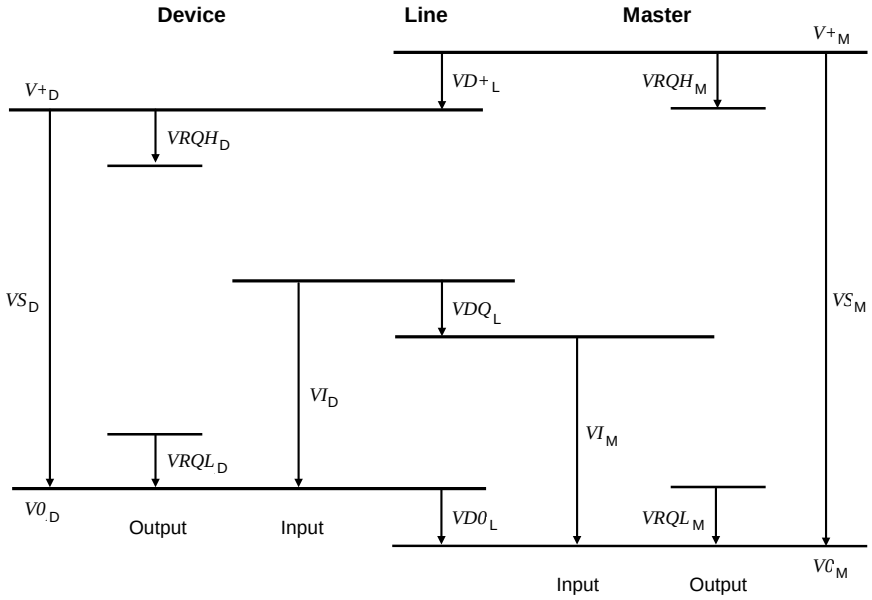


Figure 16 – Voltage level definitions

5.3.2.2 Receiver

The voltage range and switching threshold definitions are the same for Master and Device. The definitions in Table 5 apply.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 5 – Electric characteristics of a receiver

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
$VTHH_{D,M}$	Input threshold 'H'	10,5	n/a	13	V	See NOTE 1
$VTHL_{D,M}$	Input threshold 'L'	8	n/a	11,5	V	See NOTE 1
$VHYS_{D,M}$	Hysteresis between input thresholds 'H' and 'L'	0	n/a	n/a	V	Shall not be negative See NOTE 2
$VIL_{D,M}$	Permissible voltage range 'L'	$V0_{D,M}$ -1,0	n/a	n/a	V	With reference to relevant negative supply voltage
$VIH_{D,M}$	Permissible voltage range 'H'	n/a	n/a	$V+_{D,M}$ + 1,0	V	With reference to relevant positive supply voltage.
NOTE 1 Thresholds are compatible with the definitions of type 1 digital inputs in IEC 61131-2. NOTE 2 Hysteresis voltage $VHYS = VTHH - VTHL$.						

Figure 17 demonstrates the switching thresholds for the detection of Low and High signals.

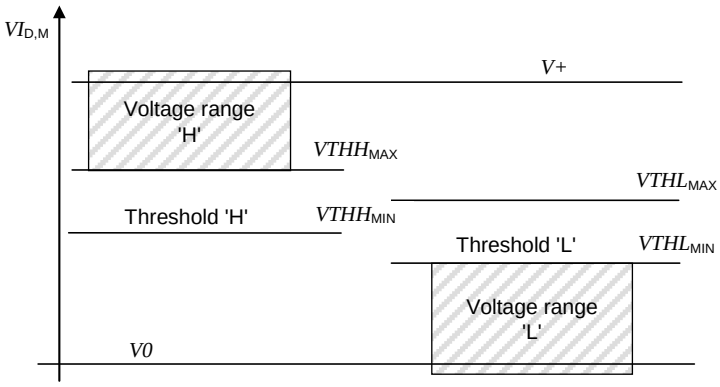


Figure 17 – Switching thresholds

5.3.2.3 Master port

The definitions in Table 6 are valid for the electric characteristics of a Master port.

Table 6 – Electric characteristics of a Master port

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
VS_M	Supply voltage for Devices	20	24	30	V	See Figure 16
IS_M	Supply current for Devices	200	n/a	n/a	mA	External supply required for > 200 mA
$ISIR_M$	Current pulse capability for Devices	400	n/a	n/a	mA	Master supply current capability for a minimum of 50 ms at 18 V after power-on of the port supply

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
ILL_M	Load or discharge current for 0 V < VI_M < 5 V 5 V < VI_M < 15 V 15 V< VI_M < 30 V	 0 5 5	 n/a n/a n/a	 15 15 15	 mA mA mA	See NOTE 1
$VRQH_M$	Residual voltage 'H'	n/a	n/a	3	V	Voltage drop relating to $V+_M$ at maximum driver current IQH_M
$VRQL_M$	Residual voltage 'L'	n/a	n/a	3	V	Voltage drop relating to $V0_M$ at maximum driver current IQL_M
IQH_M	DC driver current 'H'	100	n/a	n/a	mA	
$IQPKH_M$	Output peak current 'H'	500	n/a	n/a	mA	Absolute value See NOTE 2
IQL_M	DC driver current 'L'	100	n/a	n/a	mA	
$IQPKL_M$	Output peak current 'L'	500	n/a	n/a	mA	Absolute value See NOTE 2
CQ_M	Input capacitance	n/a	n/a	1,0	nF	f=0 MHz to 4 MHz

NOTE 1 Currents are compatible with the definition of type 1 digital inputs in IEC 61131-2. However, for the range 5 V < VI_M < 15 V, the minimum current is 5 mA instead of 2 mA in order to achieve short enough slew rates for pure p-switching Devices.

NOTE 2 Wake-up request current (5.3.3.3).

5.3.2.4 Device

The definitions in Table 7 are valid for the electric characteristics of a Device.

Table 7 – Electric characteristics of a Device

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
VS_D	Supply voltage	18	24	30	V	See Figure 16
ΔVS_D	Ripple	n/a	n/a	1,3	V_{pp}	Peak-to-peak absolute value limits shall not be exceeded. f_{ripple} = DC to 100 kHz
$VRQH_D$	Residual voltage 'H'	n/a	n/a	3	V	Voltage drop compared with $V+_D$ (IEC 60947-5-2)
$VRQL_D$	Residual voltage 'L'	n/a	n/a	3	V	Voltage drop compared with $V0_D$
IQH_D	DC driver current P-switching output ("On" state)	50	n/a	minimum ($IQPKL_M$)	mA	Minimum value due to fallback to digital input in accordance with IEC 61131-2, type 2
IQL_D	DC driver current N-switching output ("On" state)	0	n/a	minimum ($IQPKH_M$)	mA	Only for push-pull output stages
IQQ_D	Quiescent current to VO_D ("Off" state)	0	n/a	15	mA	Pull-down or residual current with deactivated output driver stages

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
CQ_D	Input capacitance	0	n/a	1,0	nF	Effective capacitance between C/Q and L+ or L- of Device in receive state

The value of 1 nF is applicable for a transmission rate of 230,4 kbit/s. Input capacitance CQ_D may be relaxed to a maximum of 10 nF in the case of push-pull stage design when operating at lower transmission rates, provided that all dynamic parameter requirements in 5.3.3.2 are met.

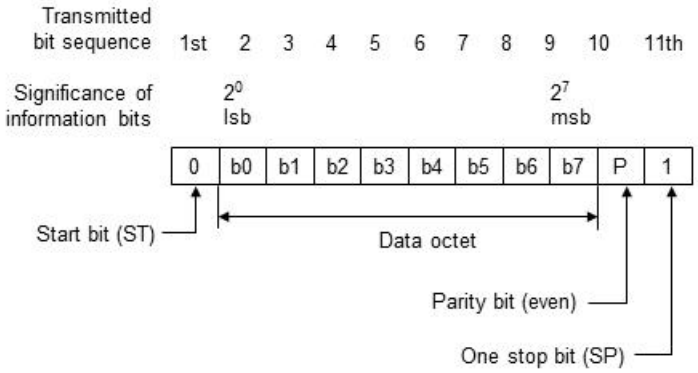
5.3.3 Timing requirements

5.3.3.1 Transmission method

The “Non Return to Zero” (NRZ) modulation is used for the bit-by-bit coding. A logic value “1” corresponds to a voltage difference of 0 V between the C/Q line and L- line. A logic value "0" corresponds to a voltage difference of +24 V between the C/Q line and L- line.

The open-circuit level on the C/Q line is 0 V with reference to L-. A start bit has logic value “0”, i.e. +24 V with reference to L-.

A UART frame is used for the "data octet"-by-"data octet" coding. The format of the SDCI UART frame is a bit string structured as shown in Figure 18.



Key:
lsb least significant bit
msb most significant bit

Figure 18 – Format of an SDCI UART frame

The definition of the UART frame format is based on ISO 1177 and ISO/IEC 2022.

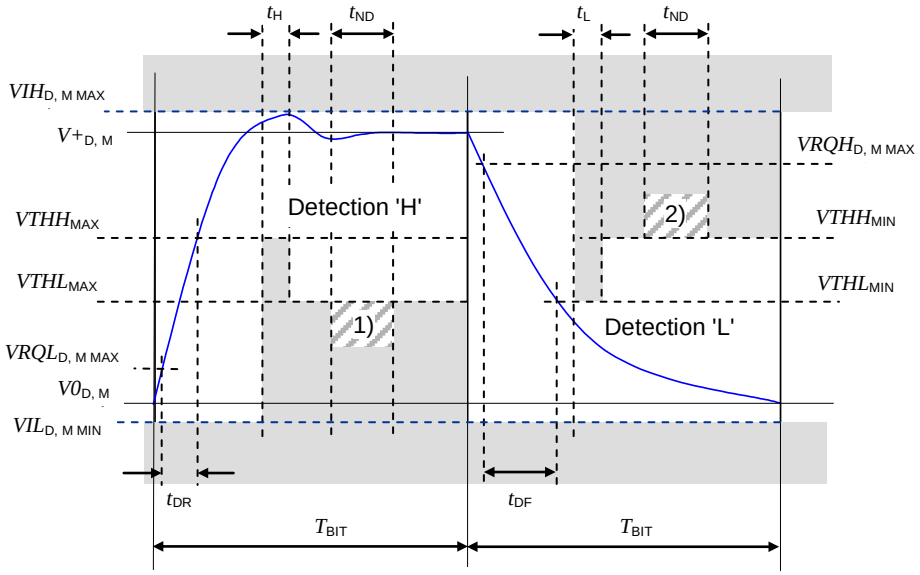
5.3.3.2 Transmission characteristics

The timing characteristics of transmission are demonstrated in the form of an eye diagram with the permissible signal ranges (see Figure 19). These ranges are applicable for receiver in both the Master and the Device.

Regardless of boundary conditions, the transmitter shall generate a voltage characteristic on the receiver's C/Q connection that is within the permissible range of the eye diagram.

The receiver shall detect bits as a valid signal shape within the permissible range of the eye diagram on the C/Q connection. Signal shapes in the “no detection” areas (below $VTHL_{MAX}$ or above $VTHH_{MIN}$ and within t_{ND}) shall not lead to invalid bits.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



NOTE In the figure, 1) = no detection "L"; and 2) = no detection "H"

Figure 19 – Eye diagram for the 'H' and 'L' detection

In order for a UART frame to be detected correctly, a signal characteristic as demonstrated in Figure 20 is required on the receiver side. The signal delay time between the C/Q signal and the UART input shall be taken into account. Time T_{BIT} always indicates the receiver's bit rate.

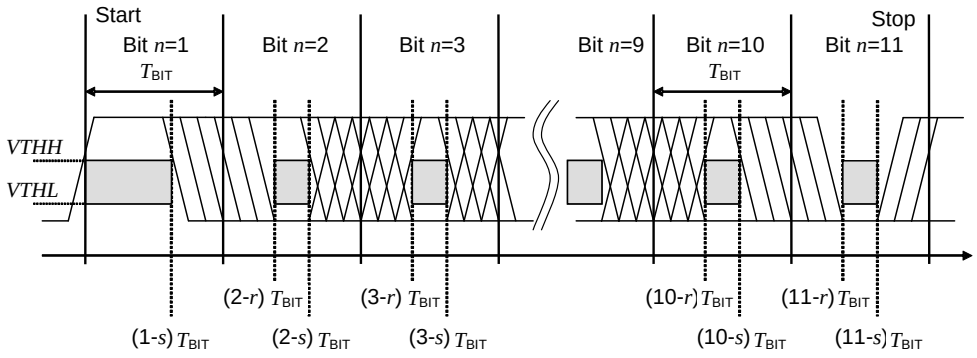


Figure 20 – Eye diagram for the correct detection of a UART frame

For every bit n in the bit sequence (n =1...11) of a UART frame, the time $(n\text{-}r)T_{BIT}$ (see Table 8 for values of r) designates the time at the end of which a correct level shall be reached in the 'H' or 'L' ranges as demonstrated in the eye diagram in Figure 19. The time (n-s) $TBIT$ (see Table 8 for values of s) describes the time, which shall elapse before the level changes. Reference shall always be made to the eye diagram in Figure 19, where signal characteristics within a bit time are concerned.

This representation permits a variable weighting of the influence parameters "transmission rate accuracy", "bit-width distortion", and "slew rate" of the receiver.

Table 8 specifies the dynamic characteristics of the transmission.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 8 – Dynamic characteristics of the transmission

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
f_{DTR}	transmission rate	n/a	4,8 38,4 230,4	n/a	kbit/s	COM1 COM2 COM3
T_{BIT}	Bit time at 4,8 kbit/s at 38,4 kbit/s at 230,4 kbit/s		 208,33 26,04 4,34		 µs µs µs	
Δf_{DTRM}	Master transmission rate accuracy at 4,8 kbit/s at 38,4 kbit/s at 230,4 kbit/s	 -0,1 -0,1 -0,1	 n/a n/a n/a	 +0,1 +0,1 +0,1	 % % %	Tolerance of the transmission rate of the Master $\Delta T_{BIT}/T_{BIT}$
r	Start of detection time within a bit with reference to the raising edge of the start bit	0,65	n/a	n/a	-	Calculated in each case from the end of a bit at a UART sampling rate of 8
s	End of detection time within a bit with reference to the raising edge of the start bit	n/a	n/a	0,22	-	Calculated in each case from the end of a bit at a UART sampling rate of 8
T_{DR}	Rise time at 4,8 kbit/s at 38,4 kbit/s at 230,4 kbit/s	0 0 0 0	n/a n/a n/a n/a	0,20 41,7 5,2 869	T_{BIT} µs µs ns	With reference to the bit time unit
t_{DF}	Fall time at 4,8 kbit/s at 38,4 kbit/s at 230,4 kbit/s	0 0 0 0	n/a n/a n/a n/a	0,20 41,7 5,2 869	T_{BIT} µs µs ns	With reference to the bit time unit
t_{ND}	Noise suppression time	n/a	n/a	1/16	T_{BIT}	Permissible duration of a receive signal above/below the detection threshold without detection taking place
t_H	Detection time High	1/16	n/a	n/a	T_{BIT}	Duration of a receive signal above the detection threshold for 'H' level
t_L	Detection time Low	1/16	n/a	n/a	T_{BIT}	Duration of a receive signal below the detection threshold for 'H' level

The parameters 'r' and 's' apply to the respective Master or Device receiver side. This definition allows for a more flexible definition of oscillator accuracy, bit distortion and slewrate on the Device side. The over-all bit-width distortion on the last bit of the UART frame shall provide a correct level in the range of Figure 20.

5.3.3.3 Wake-up current pulse

The wake-up feature is used to request that a Device goes to the COMx mode.

A service call (PL_WakeUp.req) from the DL initiates the wake-up process (see 5.2.2.2).

The wake-up request (WURQ) starts with a current pulse induced by the Master (port) for a time T_{WU}. The wake-up request comprises the following phases (see Figure 21).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

a) Injection of a current IQ_{WU} by the Master depending on the level of the C/Q connection. For an input signal equivalent to logic “1” this is a current source; for an input signal equivalent to logic “0” this is a current sink.

b) Delay time of the Device until it is ready to receive.

The wake-up request pulse can be detected by the Device through a voltage change on the C/Q line or evaluation of the current of the respective driver element within the time T_{WU}. Figure 21 shows examples for Devices with low output power.

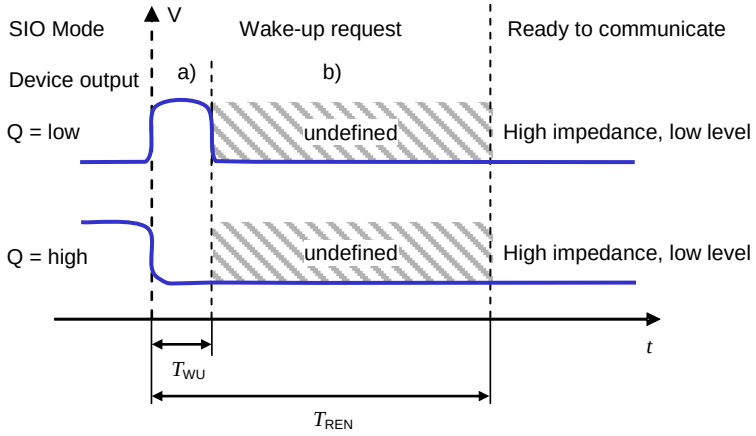


Figure 21 – Wake-up request

Table 9 specifies the current and timing properties associated with the wake-up request. See Table 6 for values of $IQPKL_M$ and $IQPKH_M$.

Table 9 – Wake-up request characteristics

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
IQ_{WU}	Amplitude of Master’s wake-up current pulse	$IQPKL_M$ or $IQPKH_M$	n/a	n/a	mA	Current pulse followed by switching status of Device
T_{WU}	Duration of Master's wake-up current pulse	75	n/a	85	µs	Master property
T_{REN}	Receive enable delay	n/a	n/a	500	µs	Device property

5.4 Power supply

5.4.1 Power supply options

The SDCI connection system provides dedicated power lines in addition to the signal line. The communication section of a Device shall always be powered by the Master using the power lines defined in the 3-wire connection system (Power1).

The maximum supply current available from a Master port is specified in Table 6.

The application part of the device may be powered in one of three ways:

- via the power lines of the SDCI 3-wire connection system (class A ports), using Power1;
- via the extra power lines of the SDCI 5-wire connection system (class B ports), using an extra power supply at the Master (Power2);

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- via a local power supply at the Device (design specific).

Port class A allows power consumption of up to 200 mA, as specified in Table 6. Maximum power consumption on port class B depends on the selected connection method. M12 only allows up to an extra 3,5 A.

5.4.2 Power-on requirements

Figure 22 shows how the power-on behavior of a Device is defined by the ramp-up time of the Power1 supply and by the Device internal time to get ready for the wake-up operation.

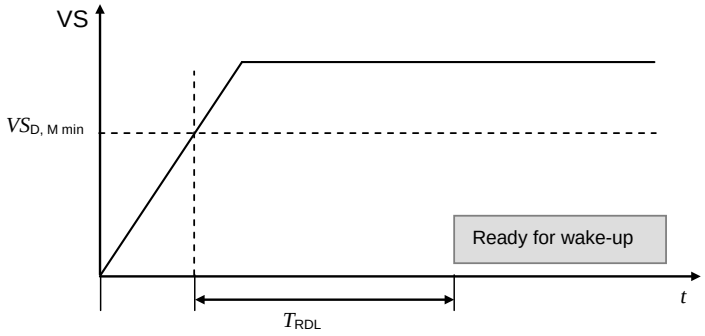


Figure 22 – Power-on timing for Power1

Upon power-on it is mandatory for a Device to reach the wake-up ready state within the time limits specified in Table 10.

Table 10 – Power-on timing

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
T_{RDL}	Wake-up readiness following power-on	n/a	n/a	300	ms	Device ramp-up time until it is ready for wake-up signal detection (See NOTE)
NOTE Equivalent to the time delay before availability in IEC 60947-5-2.						

5.5 Medium

5.5.1 Connectors

The Master and Device pin assignment is based on the specifications in IEC 60947-5-2, with extensions specified in the paragraphs below. Ports class A use M5, M8, and M12 connectors, with a maximum of four pins. Ports class B only use M12 connectors with 5 pins. M12 connectors are mechanically A-coded according to IEC 61076-2-101.

NOTE For legacy or compatibility reasons, direct wiring or different types of connectors can be used instead, provided that they do not violate the electrical characteristics and use signal naming specified in this standard.

Female connectors are assigned to the Master and male connectors to the Device. Table 11 lists the pin assignments and Figure 23 shows the layout and mechanical coding for M12, M8, and M5 connections.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 11 – Pin assignments

Pin	Signal	Designation	Remark
1	L+	Power supply (+)	See Table 7
2	I/Q P24	NC/DI/DO (port class A) P24 (port class B)	Option 1: NC (not connected) Option 2: DI Option 3: DI, then configured DO Option 4: Extra power supply for power Devices (port class B)
3	L-	Power supply (-)	See Table 7
4	C/Q	SIO/SDCI	Standard I/O mode (DI/DO) or SDCI (see Table 6 for electrical characteristics of DO).
5	NC N24	NC (port class A) N24 (port class B)	Option 1: Shall not be connected on the Master side (port class A). Option 2: Reference to the extra power supply (port class B)
NOTE M12 is always a 5 pin version on the Master side (female).			

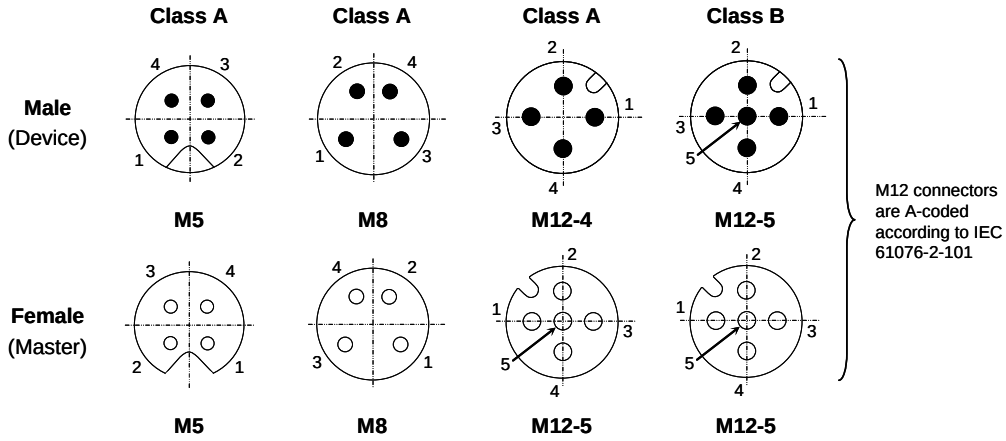


Figure 23 – Pin layout front view

Figure 24 shows the layout of the two port classes A and B. Class B ports shall be marked to distinguish them from Class A ports, because of risks deriving from incompatibilities.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

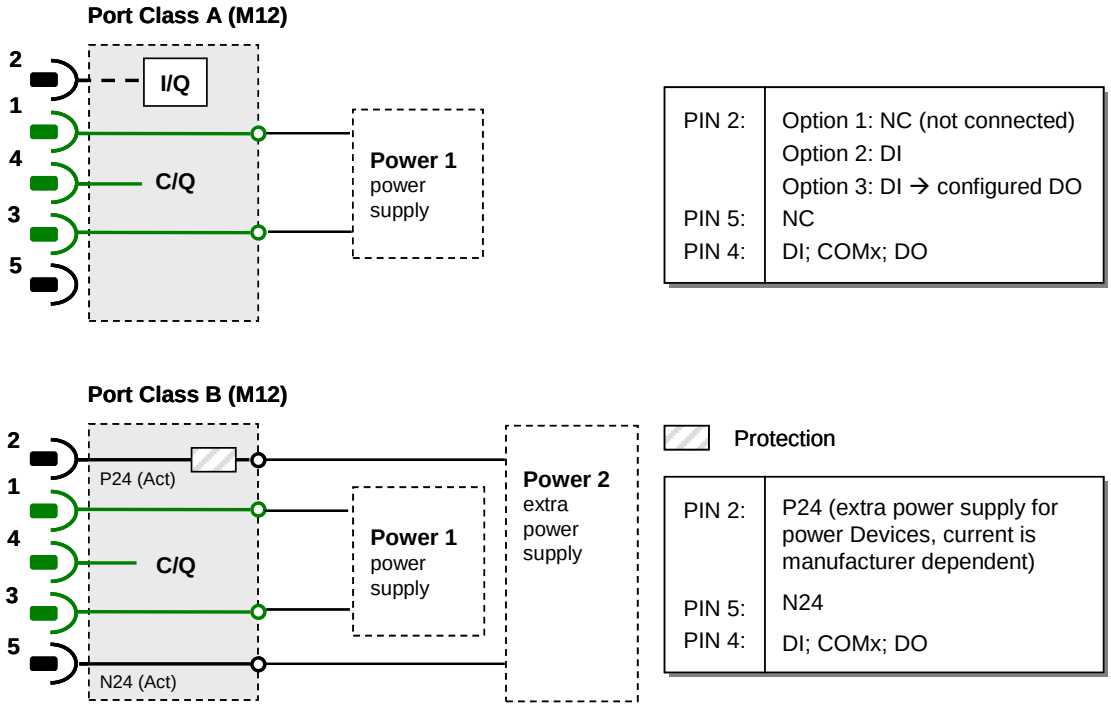


Figure 24 – Class A and B port definitions

5.5.2 Cable

The transmission medium for SDCI communication is a multi-wired cable with 3 or more wires. The definitions in the following paragraphs implicitly cover the static voltage definitions in Table 5 and Figure 16. To ensure functional reliability, the cable properties shall comply with Table 12.

Table 12 – Cable characteristics

Property	Minimum	Typical	Maximum	Unit
Length	0	n/a	20	m
Overall loop resistance RL_{eff}	n/a	n/a	6,0	Ω
Effective line capacitance CL_{eff}	n/a	n/a	3,0	nF (<1 MHz)

The loop resistance RL_{eff} and the effective line capacitance CL_{eff} may be measured as demonstrated in Figure 25.

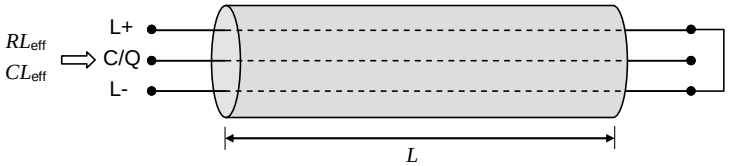


Figure 25 – Reference schematic for effective line capacitance and loop resistance

Table 13 shows the cable conductors and their assigned color codes.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 13 – Cable conductor assignments

Signal	Designation	Color	Remark
L-	Power supply (-)	Blue[a]	SDCI 3-wire connection system
C/Q	Communication signal	Black[a]	SDCI 3-wire connection system
L+	Power supply (+)	Brown[a]	SDCI 3-wire connection system
I/Q	DI or DO	White[a]	Optional
P24	Extra power supply (+)	Any other	Optional
N24	Extra power supply (-)	Any other	Optional
[a] Corresponding to IEC 60947-5-2			

6 Standard Input and Output (SIO)

Figure 83 and Figure 94 demonstrate how the SIO mode allows a Device to bypass the SDCI communication layers and to map the DI or DO signal directly into the data exchange message of the higher level fieldbus or system. Changing between the SDCI and SIO mode is defined by the user configuration or implicitly by the services of the Master applications. The system management takes care of the corresponding initialization or deactivation of the SDCI communication layers and the physical layer (mode switch). The characteristics of the interfaces for the DI and DO signals are derived from the caracteristics specified in IEC 61131-2 for type 1.

7 Data link layer (DL)

7.1 General

The data link layers of SDCI are concerned with the delivery of messages between a Master and a Device across the physical link. It uses several M-sequence ("message sequence") types for different data categories.

A set of DL-services is available to the application layer (AL) for the exchange of Process Data (PD) and On-request Data (OD). Another set of DL-services is available to system management (SM) for the retrieval of Device identification parameters and the setting of state machines within the DL. The DL uses PL-Services for controlling the physical layer (PL) and for exchanging UART frames. The DL takes care of the error detection of messages (whether internal or reported from the PL) and the appropriate remedial measures (e.g. retry).

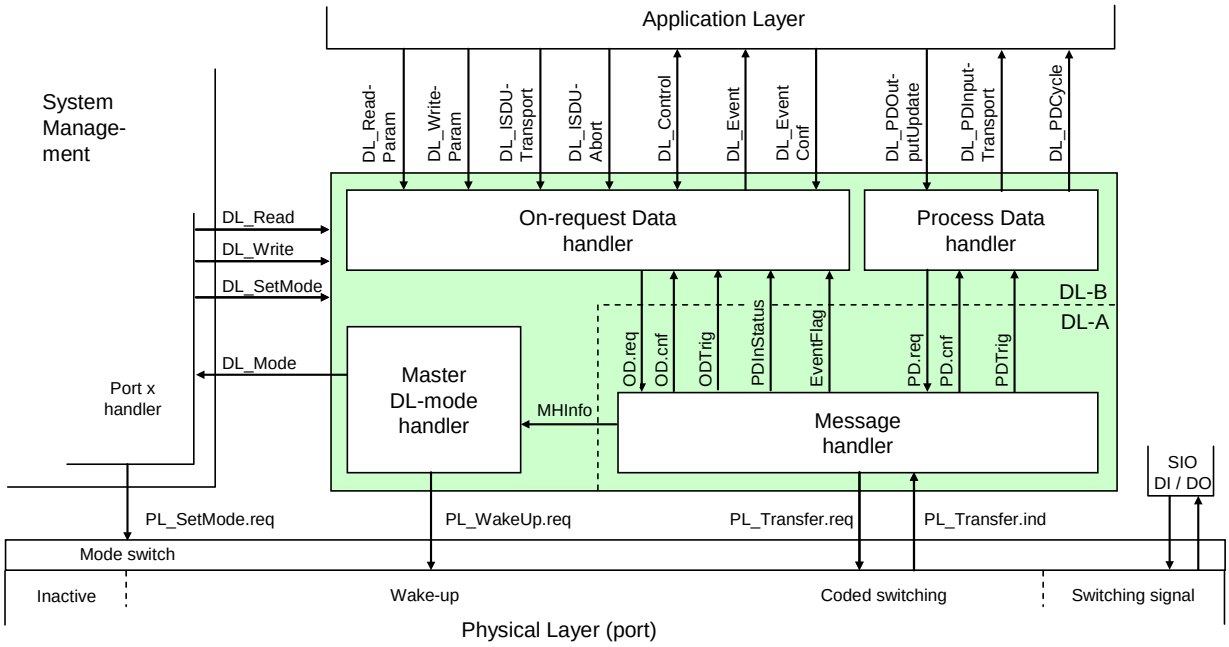
The data link layers are structured due to the nature of the data categories into Process Data handlers and On-request Data handlers which are in turn using a message handler to deal with the requested transmission of messages. The special modes of Master ports such as wake-up, COMx, and SIO (disable communication) require a dedicated DL-mode handler within the Master DL. The special wake-up signal modulation requires signal detection on the Device side and thus a DL-mode handler within the Device DL. Each handler comprises its own state machine.

The data link layer is subdivided in a DL-A section with its own internal services and a DL-B section with the external services.

The DL uses additional internal administrative calls between the handlers which are defined in the "internal items" section of the associated state-transition tables.

Figure 26 shows an overview of the structure and the services of the Master's data link layer.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



NOTE This figure uses the conventions in 3.3.5.

Figure 26 – Structure and services of the data link layer (Master)

Figure 27 shows an overview of the structure and the services of the Device's data link layer.

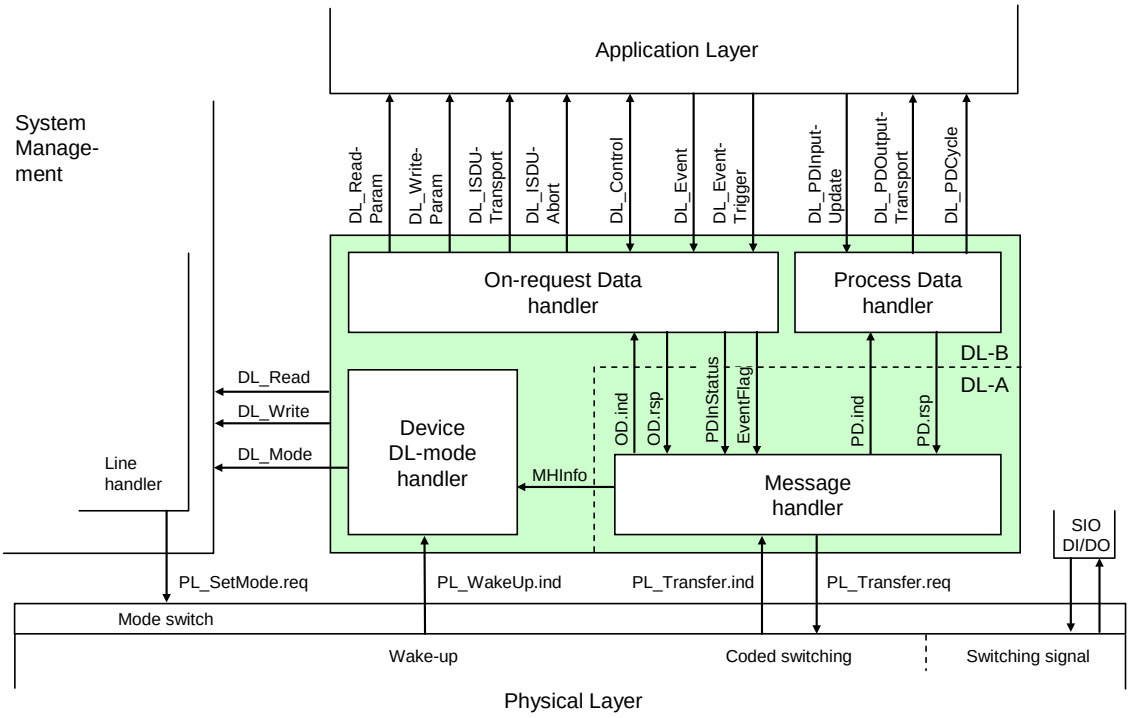


Figure 27 – Structure and services of the data link layer (Device)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2 Data link layer services

7.2.1 DL-B services

7.2.1.1 Overview of services within Master and Device

Clause 7 defines the services of the data link layer to be provided to the application layer and system management via its external interfaces. Table 14 lists the assignments of Master and Device to their roles as initiator or receiver for the individual DL services. Empty fields indicate no availability of this service on Master or Device.

Table 14 – Service assignments within Master and Device

Service name	Master	Device
DL_ReadParam	R	I
DL_WriteParam	R	I
DL_ISDUTransport	R	I
DL_ISDUAbort	R	I
DL_PDOutputUpdate	R	
DL_PDOutputTransport		I
DL_PDInputUpdate		R
DL_PDInputTransport	I	
DL_PDCycle	I	I
DL_SetMode	R	
DL_Mode	I	I
DL_Event	I	R
DL_EventConf	R	
DL_EventTrigger		R
DL_Control	I / R	R / I
DL_Read	R	I
DL_Write	R	I
Key (see 3.3.4) I Initiator of service R Receiver (responder) of service		

See 3.3 for conventions and how to read the service descriptions in 7.2, 8.2, 9.2.2, and 9.3.2.

7.2.1.2 DL_ReadParam

The DL_ReadParam service is used by the AL to read a parameter value from the Device via the page communication channel. The parameters of the service primitives are listed in Table 15.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 15 – DL_ReadParam

Parameter name	.req	.cnf	.ind
Argument	M		M
Address	M		M
Result (+)		S	
Value		M	
Result (-)		S	
ErrorInfo		M	

Argument
The service-specific parameters are transmitted in the argument.

Address
This parameter contains the address of the requested Device parameter, i.e. the Device parameter addresses within the page communication channel (see Table B.1).

Permitted values: 0 to 31

Result (+):
This selection parameter indicates that the service has been executed successfully.

Value
This parameter contains read Device parameter values.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.1.3 DL_WriteParam

The DL_WriteParam service is used by the AL to write a parameter value to the Device via the page communication channel. The parameters of the service primitives are listed in Table 16.

Table 16 – DL_WriteParam

Parameter name	.req	.cnf	.ind
Argument	M		M
Address	M		M
Value	M		M
Result (+)		S	
Result (-)		S	
ErrorInfo		M	

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Argument
The service-specific parameters are transmitted in the argument.

Address
This parameter contains the address of the requested Device parameter, i.e. the Device parameter addresses within the page communication channel.

Permitted values: 16 to 31, in accordance with Device parameter access rights

Value
This parameter contains the Device parameter value to be written.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.1.4 DL_Read

The DL_Read service is used by system management to read a Device parameter value via the page communication channel. The parameters of the service primitives are listed in Table 17.

Table 17 – DL_Read

Parameter name	.req	.cnf	.ind
Argument	M		M
Address	M		M
Result (+)		S	
Value		M	
Result (-)		S	
ErrorInfo		M	

Argument
The service-specific parameters are transmitted in the argument.

Address
This parameter contains the address of the requested Device parameter, i.e. the Device parameter addresses within the page communication channel (see Table B.1).

Permitted values: 0 to 15, in accordance with Device parameter access rights

Result (+):
This selection parameter indicates that the service has been executed successfully.

Value
This parameter contains read Device parameter values.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.1.5 DL_Write

The DL_Write service is used by system management to write a Device parameter value to the Device via the page communication channel. The parameters of the service primitives are listed in Table 18.

Table 18 – DL_Write

Parameter name	.req	.cnf	.ind
Argument	M		M
Address	M		M
Value	M		M
Result (+)		S	
Result (-)		S	
ErrorInfo		M	

Argument
The service-specific parameters are transmitted in the argument.

Address
This parameter contains the address of the requested Device parameter, i.e. the Device parameter addresses within the page communication channel.

Permitted values: 0 to 15, in accordance with parameter access rights

Value
This parameter contains the Device parameter value to be written.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.1.6 DL_ISDUTransport

The DL_ISDUTransport service is used to transport an ISDU. This service is used by the Master to send a service request from the Master application layer to the Device. It is used by

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

the Device to send a service response to the Master from the Device application layer. The parameters of the service primitives are listed in Table 19.

Table 19 – DL_ISDUTransport

Parameter name	.req	.ind	.cnf	.res
Argument	M	M		
ValueList	M	M		
Result (+)			S	S
Data			C	C
Qualifier			M	M
Result (-)			S	S
ISDUTransportErrorInfo			M	M

Argument
The service-specific parameters are transmitted in the argument.

ValueList
This parameter contains the relevant operating parameters

Parameter type: Record

Index
Permitted values: 2 to 65 535 (See B.2.1 for constraints)

Subindex
Permitted values: 0 to 255

Data
Parameter type: Octet string

Direction
Permitted values:
READ (Read operation),
WRITE (Write operation)

Result (+):
This selection parameter indicates that the service has been executed successfully.

Data
Parameter type: Octet string

Qualifier
Permitted values: an I-Service Device response according to Table A.12

Result (-):
This selection parameter indicates that the service failed.

ISDUTransportErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state),
ISDU_TIMEOUT (ISDU acknowledgement time elapsed, see Table 97),

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ISDU_NOT_SUPPORTED (ISDU not implemented),
VALUE_OUT_OF_RANGE (Service parameter value violates range definitions)

7.2.1.7 DL_ISDUAbort

The DL_ISDUAbort service aborts the current ISDU transmission. This service has no parameters. The service primitives are listed in Table 20.

Table 20 – DL_ISDUAbort

Parameter name	.req	.cnf
<none>		

The service returns with the confirmation after abortion of the ISDU transmission.

7.2.1.8 DL_PDOutputUpdate

The Master's application layer uses the DL_PDOutputUpdate service to update the output data (Process Data from Master to Device) on the data link layer. The parameters of the service primitives are listed in Table 21.

Table 21 – DL_PDOutputUpdate

Parameter name	.req	.cnf
Argument	M	
OutputData	M	
Result (+)		S
TransportStatus		M
Result (-)		S
ErrorInfo		M

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Argument
The service-specific parameters are transmitted in the argument.

OutputData
This parameter contains the Process Data provided by the application layer.

Parameter type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

TransportStatus
This parameter indicates whether the data link layer is in a state permitting data to be transferred to the communication partner(s).

Permitted values:
YES (data transmission permitted),
NO (data transmission not permitted),

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:

NO_COMM	(no communication available),
STATE_CONFLICT	(service unavailable within current state)

7.2.1.9 DL_PDOutputTransport

The data link layer on the Device uses the DL_PDOutputTransport service to transfer the content of output Process Data to the application layer (from Master to Device). The parameters of the service primitives are listed in Table 22.

Table 22 – DL_PDOutputTransport

Parameter name	.ind
Argument	M
OutputData	M

Argument
The service-specific parameters are transmitted in the argument.

OutputData
This parameter contains the Process Data to be transmitted to the application layer.

Parameter type: Octet string

7.2.1.10 DL_PDInputUpdate

The Device's application layer uses the DL_PDInputUpdate service to update the input data (Process Data from Device to Master) on the data link layer. The parameters of the service primitives are listed in Table 23.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 23 – DL_PDInputUpdate

Parameter name	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
TransportStatus		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data provided by the application layer.

Result (+):
This selection parameter indicates that the service has been executed successfully.

TransportStatus
This parameter indicates whether the data link layer is in a state permitting data to be transferred to the communication partner(s).

Permitted values:
YES (data transmission permitted),
NO (data transmission not permitted),

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.1.11 DL_PDInputTransport

The data link layer on the Master uses the DL_PDInputTransport service to transfer the content of input data (Process Data from Device to Master) to the application layer. The parameters of the service primitives are listed in Table 24.

Table 24 – DL_PDInputTransport

Parameter name	.ind
Argument	M
InputData	M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data to be transmitted to the application layer.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Parameter type: Octet string

7.2.1.12 DL_PDCycle

The data link layer uses the DL_PDCycle service to indicate the end of a Process Data cycle to the application layer. This service has no parameters. The service primitives are listed in Table 25.

Table 25 – DL_PDCycle

Parameter name	.ind
<none>	

7.2.1.13 DL_SetMode

The DL_SetMode service is used by system management to set up the data link layer's state machines and to send the characteristic values required for operation to the data link layer. The parameters of the service primitives are listed in Table 26.

Table 26 – DL_SetMode

Parameter name	.req	.cnf
Argument	M	
Mode	M	
ValueList	U	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Mode
This parameter indicates the requested mode of the Master's DL on an individual port.

Permitted values:
INACTIVE (handler shall change to the INACTIVE state),
STARTUP (handler shall change to STARTUP state),
PREOPERATE (handler shall change to PREOPERATE state),
OPERATE (handler shall change to OPERATE state)

ValueList
This parameter contains the relevant operating parameters.

Data structure: record

M-sequenceTime: (to be propagated to message handler)

M-sequenceType: (to be propagated to message handler)
Permitted values:
TYPE_0,
TYPE_1_1, TYPE_1_2, TYPE_1_V,
TYPE_2_1, TYPE_2_2, TYPE_2_3, TYPE_2_4, TYPE_2_5, TYPE_2_6, TYPE_2_V

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

(TYPE_1_1 forces interleave mode of Process and On-request Data transmission, see 7.3.4.2)

PDInputLength: (to be propagated to message handler)

PDOutputLength: (to be propagated to message handler)

OnReqDataLengthPerMessage: (to be propagated to message handler)

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (service unavailable within current state),
PARAMETER_CONFLICT (consistency of parameter set violated)

7.2.1.14 DL_Mode

The DL uses the DL_Mode service to report to system management that a certain operating status has been reached. The parameters of the service primitives are listed in Table 27.

Table 27 – DL_Mode

Parameter name	.ind
Argument	M
RealMode	M

Argument
The service-specific parameters are transmitted in the argument.

RealMode
This parameter indicates the status of the DL-mode handler.

Permitted values:
INACTIVE (Handler changed to the INACTIVE state)
COM1 (COM1 mode established)
COM2 (COM2 mode established)
COM3 (COM3 mode established)
COMLOST (Lost communication)
ESTABCOM (Handler changed to the EstablishCom state)
STARTUP (Handler changed to the STARTUP state)
PREOPERATE (Handler changed to the PREOPERATE state)
OPERATE (Handler changed to the OPERATE state)

7.2.1.15 DL_Event

The service DL_Event indicates a pending status or error information. The cause for an Event is located in a Device and the Device application triggers the Event transfer. The parameters of the service primitives are listed in Table 28.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 28 – DL_Event

Parameter name	.req	.ind
Argument	M	M
Instance	M	M
Type	M	M
Mode	M	M
EventCode	M	M
EventsLeft		M

Argument
The service-specific parameters are transmitted in the argument.

Instance
This parameter indicates the Event source.

Permitted values: Application (see Table A.17)

Type
This parameter indicates the Event category.

Permitted values: ERROR, WARNING, NOTIFICATION (see Table A.19)

Mode
This parameter indicates the Event mode.

Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table A.20)

EventCode
This parameter contains a code identifying a certain Event (see Table D.1).

Parameter type: 16 bit unsigned integer

EventsLeft
This parameter indicates the number of unprocessed Events.

7.2.1.16 DL_EventConf

The DL_EventConf service confirms the transmitted Events via the Event handler. This service has no parameters. The service primitives are listed in Table 29.

Table 29 – DL_EventConf

Parameter name	.req	.cnf
<none>		

7.2.1.17 DL_EventTrigger

The DL_EventTrigger request starts the Event signaling (see Event flag in Figure A.3) and freezes the Event memory within the DL. The confirmation is returned after the activated Events have been processed. Additional DL_EventTrigger requests are ignored until the previous one has been confirmed (see 7.3.8, 8.3.3 and Figure 64). This service has no parameters. The service primitives are listed in Table 30.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 30 – DL_EventTrigger

Parameter name	.req	.cnf
<none>		

7.2.1.18 DL_Control

The Master uses the DL_Control service to convey control information via the MasterCommand mechanism to the corresponding technology specific Device application and to get control information via the PD status flag mechanism (see A.1.5) and the PDInStatus service (see 7.2.2.5). The parameters of the service primitives are listed in Table 31.

Table 31 – DL_Control

Parameter name	.req	.ind
Argument	M	M
ControlCode	M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

ControlCode
This parameter indicates the qualifier status of the Process Data (PD)

Permitted values:

VALID	(Input Process Data valid; see 7.2.2.5, 8.2.2.12)
INVALID	(Input Process Data invalid)
PDOUTVALID	(Output Process Data valid; see 7.3.7.1)
PDOUTINVALID	(Output Process Data invalid or missing)

7.2.2 DL-A services

7.2.2.1 Overview

According to 7.1 the data link layer is split into the upper layer DL-B and the lower layer DL-A. The layer DL-A comprises the message handler as shown in Figure 26 and Figure 27.

The Master message handler encodes commands and data into messages and sends these to the connected Device via the physical layer. It receives messages from the Device via the physical layer and forwards their content to the corresponding handlers in the form of a confirmation. When the "Event flag" is set in a Device message (see A.1.5), the Master message handler invokes an EventFlag service to prompt the Event handler.

The Master message handler shall employ a retry strategy following a corrupted message, i.e. upon receiving an incorrect checksum from a Device, or no checksum at all. In these cases the Master shall repeat the Master message two times (see Table 97). If the retries are not successful, a negative confirmation shall be provided and the Master shall re-initiate the communication via the Port-x handler beginning with a wake-up.

After a start-up phase the message handler performs cyclic operation with the M-sequence type and cycle time provided by the DL_SetMode service.

Table 32 lists the assignment of Master and Device to their roles as initiator (I) or receiver (R) in the context of the execution of their individual DL-A services.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 32 – DL-A services within Master and Device

Service name	Master	Device
OD	R	I
PD	R	I
EventFlag	I	R
PDInStatus	I	R
MHInfo	I	I
ODTrig	I	
PDTrig	I	

7.2.2.2 OD

The OD service is used to set up the On-request Data for the next message to be sent. In turn, the confirmation of the service contains the data from the receiver. The parameters of the service primitives are listed in Table 33.

Table 33 – OD

Parameter name	.req	.ind	.rsp	.cnf
Argument	M	M		
RWDirection	M	M		
ComChannel	M	M		
AddressCtrl	M	M		
Length	M	M		
Data	C	C		
Result (+)			S	S
Data			C	C(=)
Length			M	M
Result (-)			S	S
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

RWDirection
This parameter indicates the read or write direction.

Permitted values:
READ (Read operation),
WRITE (Write operation)

ComChannel
This parameter indicates the selected communication channel for the transmission.

Permitted values: DIAGNOSIS, PAGE, ISDU (see Table A.1)

AddressCtrl
This parameter contains the address or flow control value (see A.1.2).

Permitted values: 0 to 31

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Length
This parameter contains the length of data to transmit.

Permitted values: 0 to 32

Data
This parameter contains the data to transmit.

Data type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

Data
This parameter contains the read data values.

Length
This parameter contains the length of the received data package.

Permitted values: 0 to 32

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:

NO_COMM	(no communication available),
STATE_CONFLICT	(service unavailable within current state)

7.2.2.3 PD

The PD service is used to setup the Process Data to be sent through the process communication channel. The confirmation of the service contains the data from the receiver. The parameters of the service primitives are listed in Table 34.

Table 34 – PD

Parameter name	.req	.ind	.rsp	.cnf
Argument	M	M		
PDInAddress	C	C(=)		
PDInLength	C	C(=)		
PDOut	C	C(=)		
PDOutAddress	C	C(=)		
PDOutLength	C	C(=)		
Result (+)			S	S
PDIn			C	C(=)
Result (-)			S	S
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

PDInAddress
This parameter contains the address of the requested input Process Data (see 7.3.4.2).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

PDInLength
This parameter contains the length of the requested input Process Data.

Permitted values: 0 to 32

PDOut
This parameter contains the Process Data to be transferred from Master to Device.

Data type: Octet string

PDOutAddress
This parameter contains the address of the transmitted output Process Data (see 7.3.4.2).

PDOutLength
This parameter contains the length of the transmitted output Process Data.

Permitted values: 0 to 32

Result (+)

This selection parameter indicates that the service has been executed successfully.

PDIn
This parameter contains the Process Data to be transferred from Device to Master.

Data type: Octet string

Result (-)

This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_COMM (no communication available),
STATE_CONFLICT (service unavailable within current state)

7.2.2.4 EventFlag

The EventFlag service sets or signals the status of the "Event flag" (see A.1.5) during cyclic communication. The parameters of the service primitives are listed in Table 35.

Table 35 – EventFlag

Parameter name	.ind	.req
Argument		
Flag	M	M

Argument
The service-specific parameters are transmitted in the argument.

Flag
This parameter contains the value of the "Event flag".

Permitted values:
TRUE ("Event flag" = 1)
FALSE ("Event flag" = 0)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.2.5 PDInStatus

The service PDInStatus sets and signals the validity qualifier of the input Process Data. The parameters of the service primitives are listed in Table 36.

Table 36 – PDInStatus

Parameter name	.req	.ind
Argument		
Status	M	M

Argument
The service-specific parameters are transmitted in the argument.

Status
This parameter contains the validity indication of the transmitted input Process Data.

Permitted values:
VALID (Input Process Data valid based on PD status flag (see A.1.5); see 7.2.1.18)
INVALID (Input Process Data invalid)

7.2.2.6 MHInfo

The service MHInfo signals an exceptional operation within the message handler. The parameters of the service are listed in Table 37.

Table 37 – MHInfo

Parameter name	.ind
Argument	
MHInfo	M

Argument
The service-specific parameters are transmitted in the argument.

MHInfo
This parameter contains the exception indication of the message handler.

Permitted values:
COMLOST (lost communication),
ILLEGAL_MESSAGETYPE (unexpected M-sequence type detected)
CHECKSUM_MISMATCH (Checksum error detected)

7.2.2.7 ODTrig

The service ODTrig is only available on the Master. The service triggers the On-request Data handler and the ISDU, Command, or Event handler currently in charge to provide the On-request Data (via the OD service) for the next Master message. The parameters of the service are listed in Table 38.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 38 – ODTrig

Parameter name	.ind
Argument	
DataLength	M

Argument
The service-specific parameters are transmitted in the argument.

DataLength
This parameter contains the available space for On-request Data (OD) per message.

7.2.2.8 PDTrig

The service PDTrig is only available on the Master. The service triggers the Process Data handler to provide the Process Data (PD) for the next Master message.

The parameters of the service are listed in Table 39.

Table 39 – PDTrig

Parameter name	.ind
Argument	
DataLength	M

Argument
The service-specific parameters are transmitted in the argument.

DataLength
This parameter contains the available space for Process Data (PD) per message.

7.3 Data link layer protocol

7.3.1 Overview

Figure 26 and Figure 27 are showing the structure of the data link layer and its components; a DL-mode handler, a message handler, a Process Data handler, and an On-request Data handler to provide the specified services. Subclauses 7.3.2 to 7.3.8 define the behaviour (dynamics) of these handlers by means of UML state machines and transition tables.

The On-request Data handler supports three independent types of data: ISDU, command and Event. Therefore, three additional state machines are working together with the On-request Data handler state machine as shown in Figure 28. Supplementary sequence or activity diagrams are demonstrating certain use cases. See IEC/TR 62390 and ISO/IEC 19505.

The elements each handler is dealing with, such as messages, wake-up procedures, interleave mode, ISDU (Indexed Service Data Units), and Events are defined within the context of the respective handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

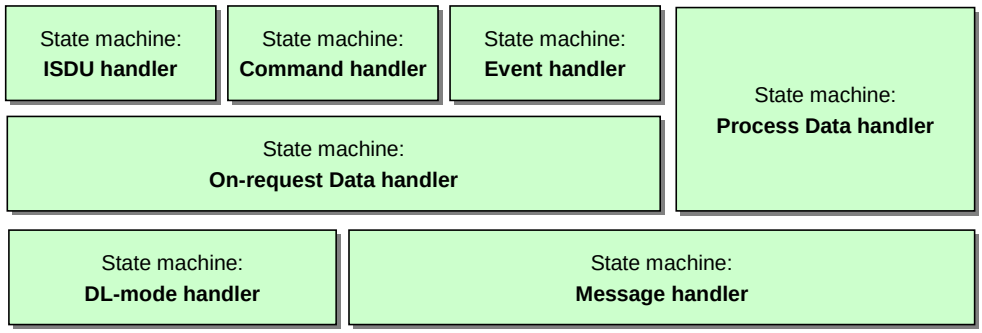


Figure 28 – State machines of the data link layer

7.3.2 DL-mode handler

7.3.2.1 General

The Master DL-mode handler shown in Figure 26 is responsible to setup the SDCI communication using services of the Physical Layer (PL) and internal administrative calls to control and monitor the message handler as well as the states of other handlers.

The Device DL-mode handler shown in Figure 27 is responsible to detect a wake-up request and to establish communication. It receives MasterCommands to synchronize with the Master DL-mode handler states STARTUP, PREOPERATE, and OPERATE and manages the activation and de-activation of handlers as appropriate.

7.3.2.2 Wake-up procedures and Device conformity rules

System management triggers the following actions on the data link layer with the help of the DL_SetMode service (requested mode = STARTUP).

The Master DL-mode handler tries to establish communication via a wake-up request (PL_WakeUp.req) followed by a test message with M-sequence TYPE_0 (read "MinCycleTime") according to the sequence shown in Figure 29.

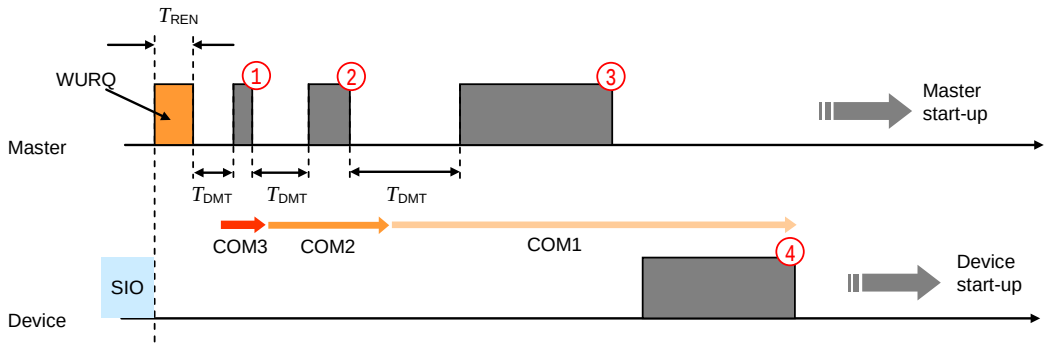


Figure 29 – Example of an attempt to establish communication

After the wake-up request (WURQ), specified in 5.3.3.3, the DL-mode handler requests the message handler to send the first test message after a time T_{REN} (see Table 9) and T_{DMT} (see Table 40). The specified transmission rates of COM1, COM2, and COM3 are used in descending order until a response is obtained, as shown in the example of Figure 29:

Step ①: Master message with transmission rate of COM3 (see Table 8).

Step ②: Master message with transmission rate of COM2 (see Table 8).

Step ③: Master message with transmission rate of COM1 (see Table 8).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Step ④: Device response message with transmission rate of COM1.

Before initiating a (new) message, the DL-mode handler shall wait at least for a time of T_{DMT}. T_{DMT} is specified in Table 40.

The following conformity rule applies for Devices regarding support of transmission rates:

- a Device shall support only one of the transmission rates of COM1, COM2, or COM3.

If an attempt to establish communication fails, the Master DL-mode handler shall not start a new retry wake-up procedure until after a time T_{DWU} as shown in Figure 30 and specified in Table 40.

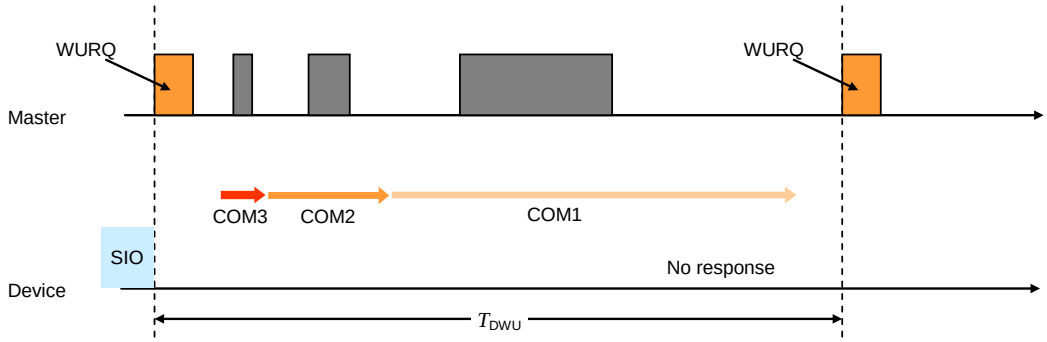


Figure 30 – Failed attempt to establish communication

The Master shall make up to n_{WU}+1 successive wake-up requests as shown in Figure 31. If this initial wake-up retry sequence fails, the Device shall reset its C/Q line to SIO mode after a time T_{DSIO} (T_{DSIO} is retrigged in the Device after each detected WURQ). The Master shall not trigger a new wake-up retry sequence until after a time T_{SD}.

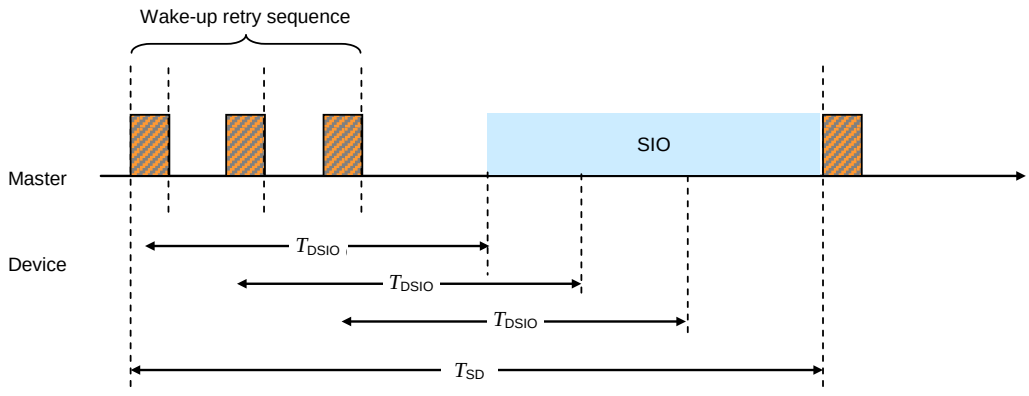


Figure 31 – Retry strategy to establish communication

The DL of the Master shall request the PL to go to SIO mode after a failed wake-up retry sequence.

The values for the timings of the wake-up procedures and retries are specified in Table 9 and Table 40. They are defined from a Master's point of view.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 40 – Wake-up procedure and retry characteristics

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
T_{DMT}	Master message delay	27	n/a	37	T_{BIT}	Bit time of subsequent data transmission rate
T_{DSIO}	Standard IO delay	60	n/a	300	ms	After T_{DSIO} the Device falls back to SIO mode (if supported)
T_{DWU}	Wake-up retry delay	30	n/a	50	ms	After T_{DWU} the Master repeats the wake-up request
n_{WU}	Wake-up retry count	2	2	2		Number of wake-up request retries
T_{SD}	Device detection time	0,5	n/a	1	s	Time between 2 wake-up request sequences. (See NOTE)
NOTE Characteristic of the Master.						

The Master's data link layer shall stop the establishing communication procedure once it finds a communicating Device, and shall report the detected COMx-Mode to system management using a DL_Mode indication. If the procedure fails, a corresponding error is reported using the same service.

7.3.2.3 Fallback procedure

System management induces the following actions on the data link layer with the help of the DL_SetMode service (mode = INACTIVE).

- A MasterCommand "Fallback" (see Table B.2) forces the Device to change to the SIO mode.
- The Device shall accomplish the transition to the SIO mode after 3 MasterCycleTimes and/or within 500 ms after the MasterCommand "Fallback". This allows for possible retries if the MasterCommand failed indicated through a negative Device response.

Figure 32 shows the fallback procedure and its retry and timing constraints.

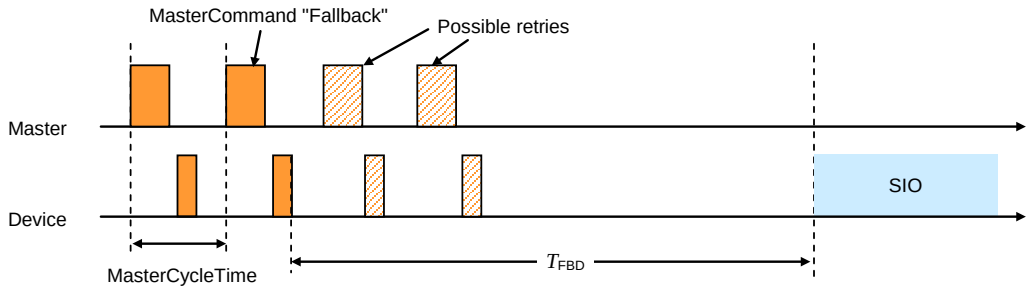


Figure 32 – Fallback procedure

Table 41 specifies the fallback timing characteristics. See A.2.6 for details.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 41 – Fallback timing characteristics

Property	Designation	Minimum	Typical	Maximum	Unit	Remark
T_{FBD}	Fallback delay	3 MasterCycle-Times (OPERATE) or 3 $T_{initcyc}$ (PREOPERATE)	n/a	500	ms	After a time T_{FBD} the Device shall be switched to SIO mode (see Figure 32)

7.3.2.4 State machine of the Master DL-mode handler

Figure 33 shows the state machine of the Master DL-mode handler.

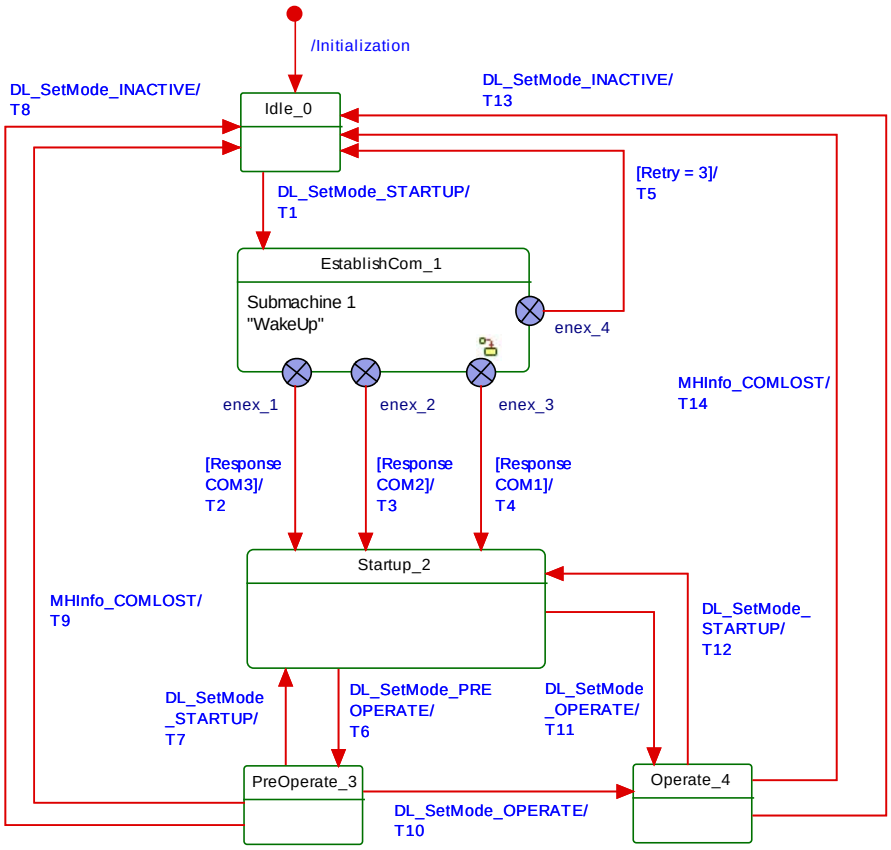


Figure 33 – State machine of the Master DL-mode handler

NOTE The conventions of the UML diagram types are defined in 3.3.7.

After reception of the service DL_SetMode_STARTUP from system management, the DL-mode handler shall first create a wake-up current pulse via the PL_WakeUp service and then establish communication. This procedure is specified in submachine 1 in Figure 34.

The purpose of state "Startup_2" is to check a Device's identity via the data of the Direct Parameter page (see Figure 5). In state "PreOperate_3", the Master assigns parameters to the Device using ISDUs. Cyclic exchange of Process Data is performed in state "Operate". Within this state additional On-request Data such as ISDUs, commands, and Events can be transmitted using appropriate M-sequence types (see Figure 37).

In state PreOperate_3 and Operate_4 different sets of handlers within the Master are activated.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

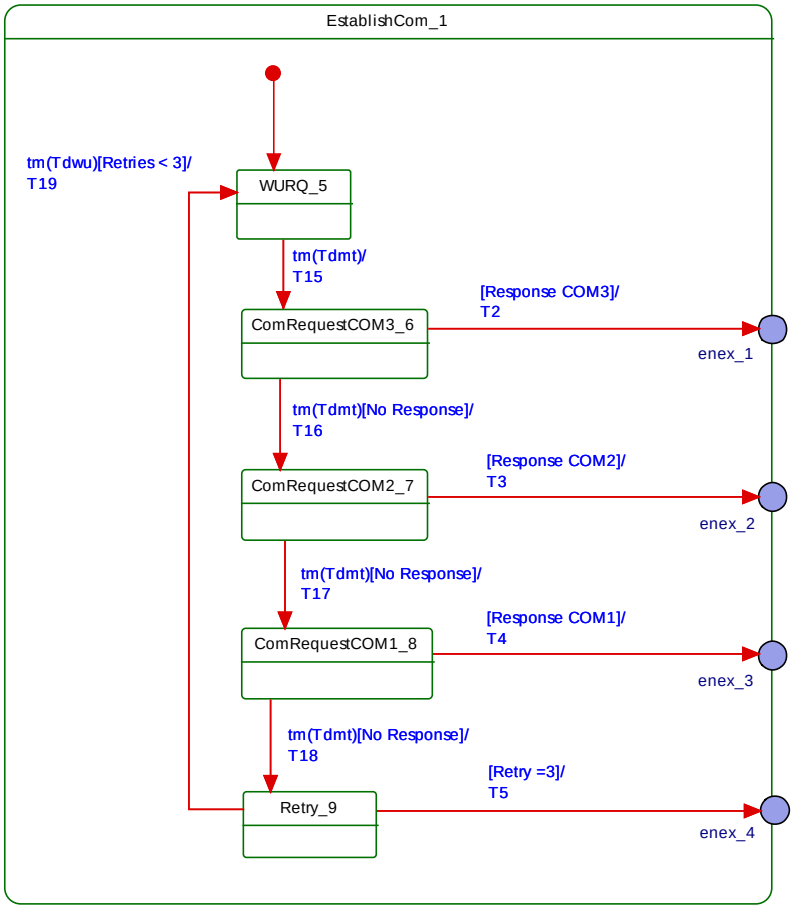


Figure 34 – Submachine 1 to establish communication

Table 42 shows the state transition tables of the Master DL-mode handler.

Table 42 – State transition tables of the Master DL-mode handler

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting on wakeup request from System Management (SM): DL_SetMode (STARTUP)
EstablishComm_1	Perform wakeup procedure (submachine 1)
Startup_2	System Management uses the STARTUP state for Device identification, check, and communication configuration (see Figure 69)
Preoperate_3	On-request Data exchange (parameter, commands, Events) without Process Data
Operate_4	Process Data and On-request Data exchange (parameter, commands, Events)
SM: WURQ_5	Create wakeup current pulse: Invoke service PL-Wake-Up (see Figure 11 and 5.3.3.3) and wait T_{DMT} (see Table 40).
SM: ComRequestCOM3_6	Try test message with transmission rate of COM3 via the message handler: Call MH_Conf_COMx (see Figure 38) and wait T_{DMT} (see Table 40).
SM: ComRequestCOM2_7	Try test message with transmission rate of COM2 via the message handler: Call MH_Conf_COMx (see Figure 38) and wait T_{DMT} (see Table 40).
SM: ComRequestCOM1_8	Try test message with transmission rate of COM1 via the message handler: Call MH_Conf_COMx (see Figure 38) and wait T_{DMT} (see Table 40).
SM: Retry_9	Check number of Retries

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Set Retry = 0.
T2	1	2	Transmission rate of COM3 successful. Message handler activated and configured to COM3 (see Figure 38, Transition T2). Activate command handler (call CH_Conf_ACTIVE in Figure 51). Return DL_Mode.ind (STARTUP) and DL_Mode.ind (COM3) to SM.
T3	1	2	Transmission rate of COM2 successful. Message handler activated and configured to COM2 (see Figure 38, Transition T2). Activate command handler (call CH_Conf_ACTIVE in Figure 51). Return DL_Mode.ind (STARTUP) and DL_Mode.ind (COM2) to SM.
T4	1	2	Transmission rate of COM1 successful. Message handler activated and configured to COM1 (see Figure 38, Transition T2). Activate command handler (call CH_Conf_ACTIVE in Figure 51). Return DL_Mode.ind (STARTUP) and DL_Mode.ind (COM1) to SM.
T5	1	0	Return DL_Mode.ind (INACTIVE) to SM.
T6	2	3	SM requested the PREOPERATE state. Activate On-request Data (call OH_Conf_ACTIVE in Figure 46), ISDU (call IH_Conf_ACTIVE in Figure 49), and Event handler (call EH_Conf_ACTIVE in Figure 53). Change message handler state to PREOPERATE (call MH_Conf_PREOPERATE in Figure 38). Return DL_Mode.ind (PREOPERATE) to SM.
T7	3	2	SM requested the STARTUP state. Change message handler state to STARTUP (call MH_Conf_STARTUP in Figure 38). Deactivate On-request Data (call OH_Conf_INACTIVE in Figure 46), ISDU (call IH_Conf_INACTIVE in Figure 49), command (call CH_Conf_INACTIVE in Figure 51) and Event handler (call EH_Conf_INACTIVE in Figure 53). Return DL_Mode.ind (STARTUP) to SM.
T8	3	0	SM requested the SIO mode. Deactivate all handlers (call xx_Conf_INACTIVE). Return DL_Mode.ind (INACTIVE) to SM.
T9	3	0	Message handler informs about lost communication via the DL-A service MHInfo (COMLOST). Deactivate all handlers (call xx_Conf_INACTIVE). Return DL_Mode.ind (COMLOST) to SM.
T10	3	4	SM requested the OPERATE state. Activate the Process Data handler (call PD_Conf_SINGLE if M-sequence type = TYPE_2_x, or PD_Conf_INTERLEAVE if M-sequence type = TYPE_1_1 in Figure 44). Change message handler state to OPERATE (call MH_Conf_OPERATE in Figure 38). Return DL_Mode.ind (OPERATE) to SM.
T11	2	4	SM requested the OPERATE state. Activate the Process Data handler (call PD_Conf_SINGLE or PD_Conf_INTERLEAVE in Figure 44 according to the Master port configuration). Activate On-request Data (call OH_Conf_ACTIVE in Figure 46), ISDU (call IH_Conf_ACTIVE in Figure 49), and Event handler (call EH_Conf_ACTIVE in Figure 53). Change message handler state to OPERATE (call MH_Conf_OPERATE in Figure 38). Return DL_Mode.ind (OPERATE) to SM.
T12	4	2	SM requested the STARTUP state. Change message handler state to STARTUP (call MH_Conf_STARTUP in Figure 38). Deactivate Process Data (call PD_Conf_INACTIVE in Figure 44), On-request Data (call OH_Conf_INACTIVE in Figure 46), ISDU (call IH_Conf_INACTIVE in Figure 49), and Event handler (call EH_Conf_INACTIVE in Figure 53). Return DL_Mode.ind (STARTUP) to SM.
T13	4	0	SM requested the SIO state. Deactivate all handlers (call xx_Conf_INACTIVE). Return DL_Mode.ind (INACTIVE) to SM.
T14	4	0	Message handler informs about lost communication via the DL-A service MHInfo (COMLOST). Deactivate all handlers (call xx_Conf_INACTIVE). Return DL_Mode.ind (COMLOST) to SM.
T15	5	6	Set transmission rate of COM3 mode.
T16	6	7	Set transmission rate of COM2 mode.
T17	7	8	Set transmission rate of COM1 mode.
T18	8	9	Increment Retry
T19	9	5	-

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
MH_Conf_COMx	Call	This call causes the message handler to send a message with the requested transmission rate of COMx and with M-sequence TYPE_0 (see Table 44).
MH_Conf_STARTUP	Call	This call causes the message handler to switch to the STARTUP state (see Figure 38)
MH_Conf_PREOPERATE	Call	This call causes the message handler to switch to the PREOPERATE state (see Figure 38)
MH_Conf_OPERATE	Call	This call causes the message handler to switch to the OPERATE state (see Figure 38)
xx_Conf_ACTIVE	Call	This call activates the respective handler. xx is substitute for MH (message handler), OH (On-request Data handler), IH (ISDU handler), CH (Command handler), and/or EH (Event handler)
xx_Conf_INACTIVE	Call	This call deactivates the message handler. xx is substitute for MH (message handler), OH (On-request Data handler), IH (ISDU handler), CH (Command handler), and/or EH (Eventhandler)
Retry	Variable	Number of retries to establish communication

7.3.2.5 State machine of the Device DL-mode handler

Figure 35 shows the state machine of the Device DL-mode handler. In state PreOperate_3 and Operate_4 different sets of handlers within the Device are activated.

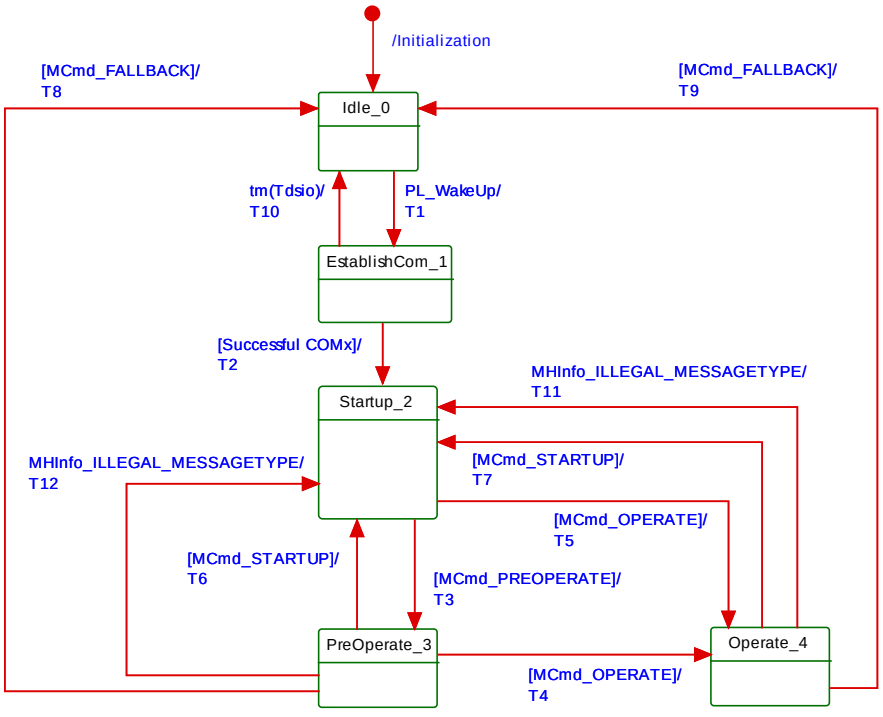


Figure 35 – State machine of the Device DL-mode handler

The Master uses MasterCommands (see Table 42) to change the Device to SIO, STARTUP, PREOPERATE, and OPERATE states. Whenever the message handler detects illegal (unexpected) M-sequence types, it will cause the DL-mode handler to change to the STARTUP state and to indicate this state to its system mangement (see 9.3.3.2) for the purpose of synchronization of Master and Device.

Table 43 shows the state transition tables of the Device DL-mode handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 43 – State transition tables of the Device DL-mode handler

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting on a detected wakeup current pulse (PL_WakeUp.ind).
EstablishComm_1	Message handler activated and waiting for the COMx test messages (see Table 42)
Startup_2	Compatibility check (see 9.2.3.3)
Preoperate_3	On-request Data exchange (parameter, commands, Events) without Process Data
Operate_4	Process Data (PD) and On-request Data exchange (parameter, commands, Events)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Wakeup current pulse detected. Activate message handler (call MH_Conf_ACTIVE in Figure 42). Indicate state via service DL_Mode.ind (ESTABCOM) to SM.
T2	1	2	One out of the three transmission rates of COM3, COM2, or COM1 mode established. Activate On-request Data (call OH_Conf_ACTIVE in Figure 47) and command handler (call CH_Conf_ACTIVE in Figure 52). Indicate state via service DL_Mode.ind (COM1, COM2, or COM3) to SM.
T3	2	3	Device command handler received MasterCommand (MCmd_PREOPERATE). Activate ISDU (call IH_Conf_ACTIVE in Figure 50) and Event handler (call EH_Conf_ACTIVE in Figure 54). Indicate state via service DL_Mode.ind (PREOPERATE) to SM.
T4	3	4	Device command handler received MasterCommand (MCmd_OPERATE). Activate Process Data handler (call PD_Conf_ACTIVE in Figure 45). Indicate state via service DL_Mode.ind (OPERATE) to SM.
T5	2	4	Device command handler received MasterCommand (MCmd_OPERATE). Activate Process Data handler (call PD_Conf_ACTIVE in Figure 45), ISDU (call IH_Conf_ACTIVE in Figure 50), and Event handler (call EH_Conf_ACTIVE in Figure 54). Indicate state via service DL_Mode.ind (OPERATE) to SM.
T6	3	2	Device command handler received MasterCommand (MCmd_STARTUP). Deactivate ISDU (call IH_Conf_INACTIVE in Figure 50) and Event handler (call EH_Conf_INACTIVE in Figure 54). Indicate state via service DL_Mode.ind (STARTUP) to SM.
T7	4	2	Device command handler received MasterCommand (MCmd_STARTUP). Deactivate Process Data handler (call PD_Conf_INACTIVE in Figure 45), ISDU (call IH_Conf_INACTIVE in Figure 50), and Event handler (call EH_Conf_INACTIVE in Figure 54). Indicate state via service DL_Mode.ind (STARTUP) to SM.
T8	3	0	Device command handler received MasterCommand (MCmd_FALLBACK). Wait until T_{FBD} elapsed, and then deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM (see Figure 79 and Table 93).
T9	4	0	Device command handler received MasterCommand (MCmd_FALLBACK). Wait until T_{FBD} elapsed, and then deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM (see Figure 79 and Table 93).
T10	1	0	After unsuccessful wakeup procedures (see Figure 30) the Device establishes the configured SIO mode after an elapsed time T_{DSIO} (see Figure 31). Deactivate all handlers (call xx_Conf_INACTIVE). Indicate state via service DL_Mode.ind (INACTIVE) to SM.
T11	4	2	Message handler detected an illegal M-sequence type. Deactivate Process Data (call PD_Conf_INACTIVE in Figure 45), ISDU (call IH_Conf_INACTIVE in Figure 50), and Event handler (call EH_Conf_INACTIVE in Figure 54). Indicate state via service DL_Mode.ind (STARTUP) to SM (see Figure 79 and Table 93).
T12	3	2	Message handler detected an illegal M-sequence type. Deactivate ISDU (call IH_Conf_INACTIVE in Figure 50) and Event handler (call EH_Conf_INACTIVE in Figure 54). Indicate state via service DL_Mode.ind (STARTUP) to SM (see Figure 79 and Table 93).

INTERNAL ITEMS	TYPE	DEFINITION
T_{FBD}	Time	See Table 41.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
T_{DSIO}	Time	See Figure 31
MCmd_XXXXXXX	Call	Any MasterCommand received by the Device command handler (see Table 42 and Figure 52, state "CommandHandler_2")

7.3.3 Message handler

7.3.3.1 General

The role of the message handler is specified in 7.1 and 7.2.2.1. Subclause 7.3.3 specifies the structure and types of M-sequences and the behaviour (dynamics) of the message handler.

7.3.3.2 M-sequences

A Master and its Device exchange data by means of a sequence of messages (M-sequence). An M-sequence comprises a message from the Master followed by a message from the Device as shown in Figure 36. Each message consists of UART frames.

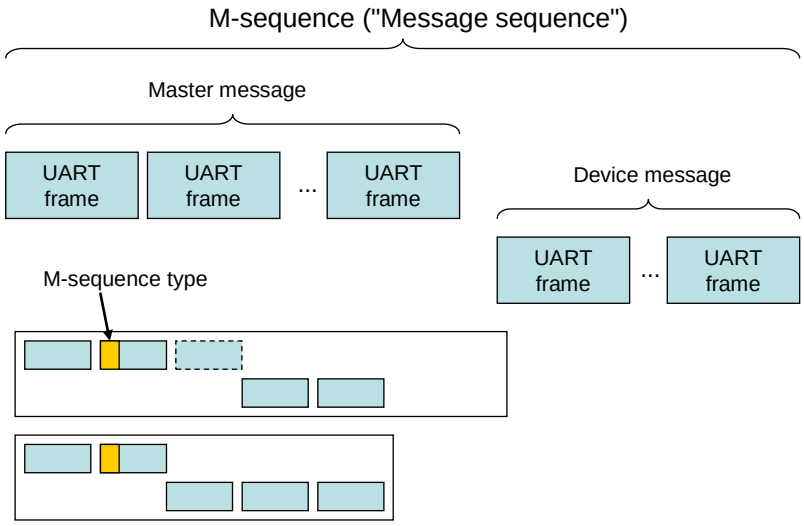


Figure 36 – SDCI message sequences

All the multi-octet data types shall be transmitted as a big-endian sequence, i.e. the most significant octet (MSO) shall be sent first, followed by less significant octets in descending order, with the least significant octet (LSO) being sent last, as shown in Figure 2.

The Master message starts with the "M-sequence Control" (MC) octet, followed by the "CHECK/TYPE" (CKT) octet, and optionally followed by either "Process Data" (PD) and/or "On-request Data" (OD) octets. The Device message in turn starts optionally with "Process Data" (PD) octets and/or "On-request Data" (OD) octets, followed by the "CHECK/STAT" (CKS) octet.

Various M-sequence types can be selected to meet the particular needs of an actuator or sensor (scan rate, amount of Process Data). The length of Master and Device messages may vary depending on the type of messages and the data transmission direction, see Figure 36.

Figure 37 presents an overview of the defined M-sequence types. Parts within dotted lines depend on the read or write direction within the M-sequence control octet.

The fixed M-sequence types consist of TYPE_0, TYPE_1_1, TYPE_1_2, and TYPE_2_1 through TYPE_2_6. The variable M-sequence types consist of TYPE_1_V and TYPE_2_V.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The different M-sequence types meet the various requirements of sensors and actuators regarding their Process Data width and respective conditions. See Clause A.2 for details of M-sequence types. See Clause A.3 for the timing constraints with M-sequences.

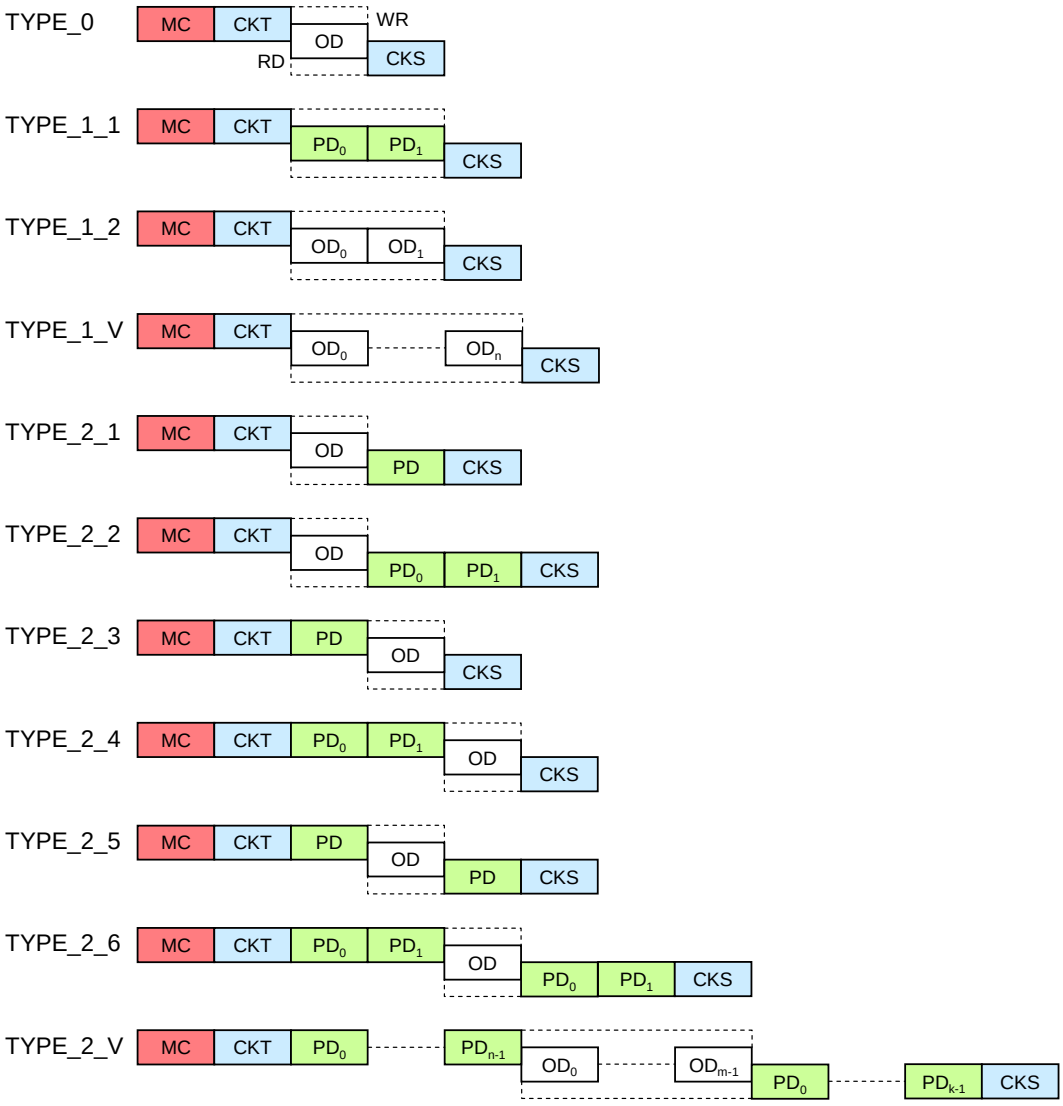


Figure 37 – Overview of M-sequence types

7.3.3.3 MasterCycleTime constraints

Within state STARTUP and PREOPERATE a Device is able to communicate in an acyclic manner. In order to detect the disconnecting of Devices it is highly recommended for the Master to perform from this point on a periodic communication ("keep-alive message") via acyclic M-sequences through the data link layer. The minimum recovery times for acyclic communication specified in A.2.6 shall be considered.

After these phases, cyclic Process Data communication can be started by the Master via the DL_SetMode (OPERATE) service. M-sequence types for the cyclic data exchange shall be used in this communication phase to exchange Process Data (PD) and On-request Data with a Device (see Table A.9 and Table A.10).

The Master shall use for time t_{CYC} the value indicated in the Device parameter "MasterCycleTime" (see Table B.1) with a relative tolerance of 0 % to +10 % (including jitter).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

In cases, where a Device has to be switched back to SIO mode after parameterization, the Master shall send a command "Fallback" (see Table B.2), which is followed by a confirmation from the Device.

7.3.3.4 State machine of the Master message handler

Figure 38 shows the Master state machine of the Master message handler. Three submachines describing reactions on communication errors are shown in Figure 39, Figure 40, and Figure 41.

The message handler takes care of the special communication requirements within the states "EstablishCom", "Startup", "PreOperate", and "Operate" of the DL-Mode handler.

An internal administrative call MH_Conf_COMx in state "Inactive_0" causes the message handler to send "test" messages with M-sequence TYPE_0 and different transmission rates of COM3, COM2, or COM1 during the establish communication sequence.

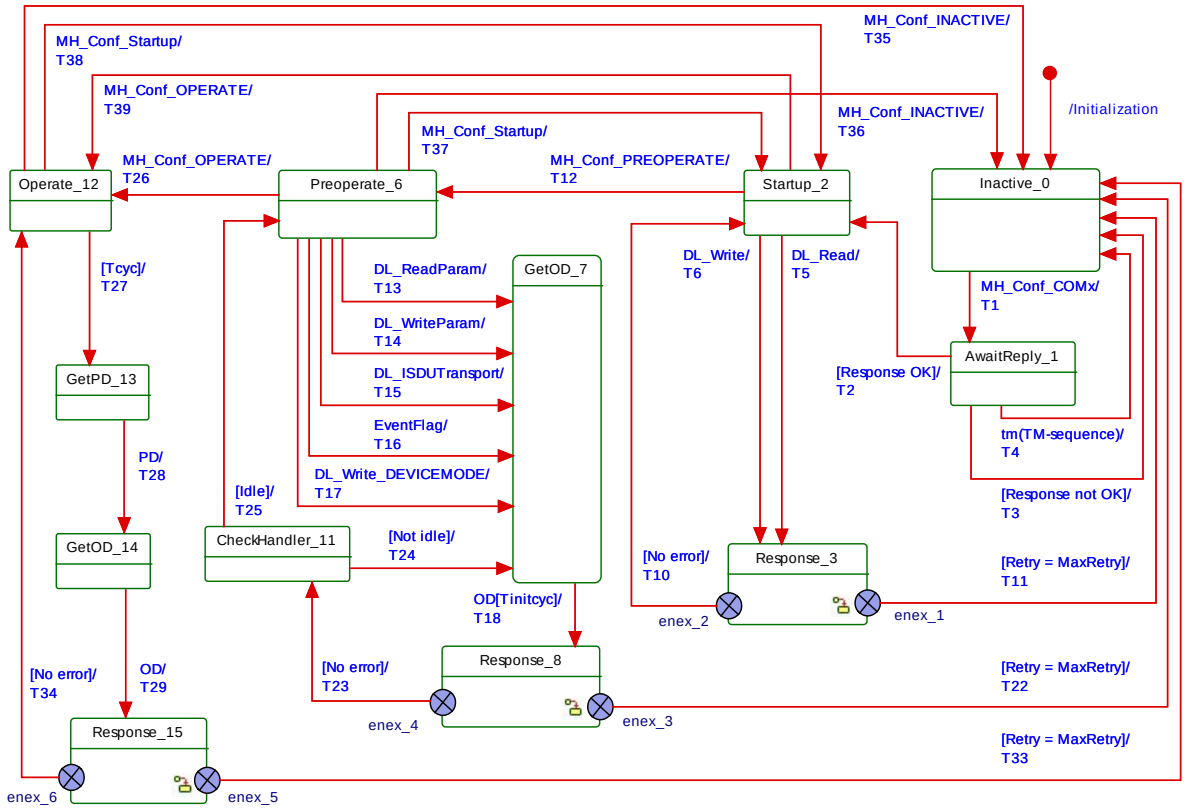


Figure 38 – State machine of the Master message handler

The state "Startup_2" provides all the communication means to support the identity checks of system management with the help of DL_Read and DL_Write services. The message handler waits on the occurrence of these services to send and receive messages (acyclic communication).

The state "Preoperate_6" is the checkpoint for all On-request Data activities such as ISDUs, commands, and Events for parameterization of the Device. The message handler waits on the occurrence of the services shown in Figure 38 to send and receive messages (acyclic communication).

The state "Operate_12" is the checkpoint for cyclic Process Data exchange. Depending on the M-sequence type the message handler generates Master messages with Process Data

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

acquired from the Process Data handler via the PD service and optionally On-request Data acquired from the On-request Data handler via the OD service.

Figure 39 shows the submachine of state "Response 3".

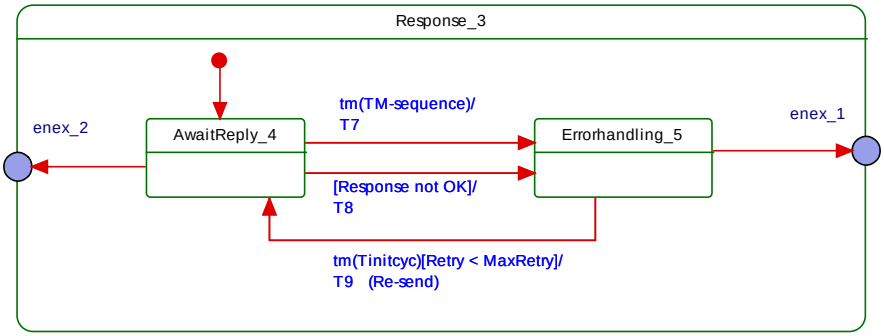


Figure 39 – Submachine "Response 3" of the message handler

Figure 40 shows the submachine of state "Response 8".

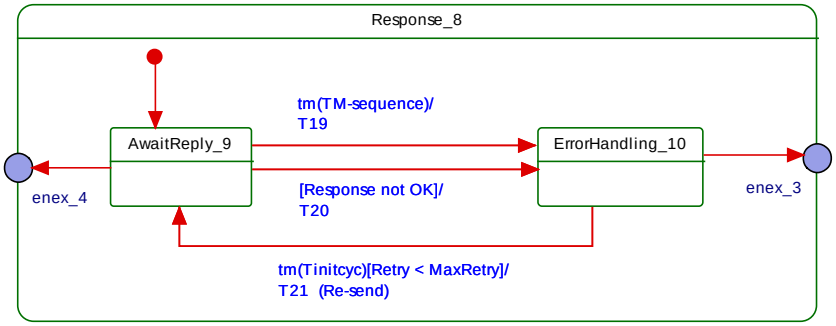


Figure 40 – Submachine "Response 8" of the message handler

Figure 41 shows the submachine of state "Response 15".

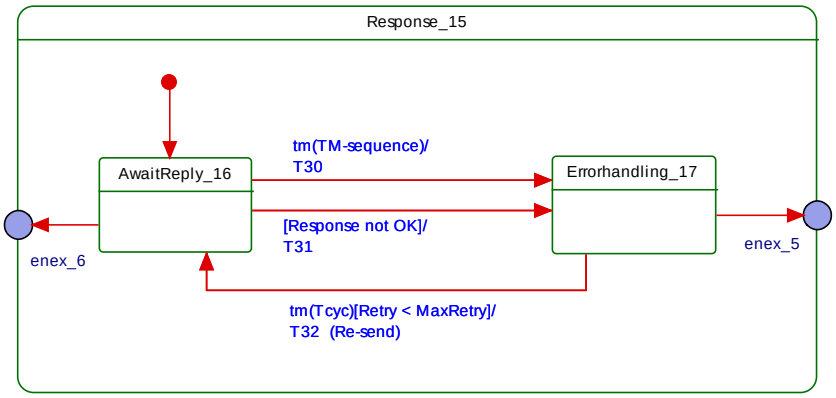


Figure 41 – Submachine "Response 15" of the message handler

Table 44 shows the state transition tables of the Master message handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 44 – State transition table of the Master message handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on demand for a "test" message via MH_Conf_COMx call (see Figure 34 and Table 42) from DL-mode handler.
AwaitReply_1	Waiting on response from the Device to the "test" message. Return to Inactive_0 state whenever the time $T_{M\text{-}sequence}$ elapsed without response from the Device or the response to the "test" message could not be decoded. In case of a correct response from the Device, the message handler changes to the Startup_2 state.
Startup_2	When entered via transition T2, this state is responsible to control acyclic On-request Data exchange according to conditions specified in Table A.7. Any service DL_Write or DL_Read from system management causes a transition.
Response_3	The OD service caused the message handler to send a corresponding message.The submachine in this pseudo state waits on the response and checks its correctness.
SM: AwaitReply_4	This state checks whether the time $T_{M\text{-}sequence}$ elapsed and the response is correct.
SM: ErrorHandling_5	In case of an incorrect response the message handler will re-send the message after a waiting time $T_{initcyc}$. After too many retries the message handler will change to the Inactive_0 state.
Preoperate_6	Upon reception of a call MH_Conf_PREOPERATE the message handler changed to this state. The message handler is now responsible to control acyclic On-request Data exchange according to conditions specified in Table A.8. Any service DL_ReadParam, DL_WriteParam, DL_ISDUTransport, DL_Write, or EventFlag causes a transition.
GetOD_7	The message handler used the ODTrig service to aquire OD from the On-request Data handler. The message handler waits on the OD service to send a message after a time $T_{initcyc}$.
Response_8	The OD service caused the message handler to send a corresponding message.The submachine in this pseudo state waits on the response and checks its correctness.
SM: AwaitReply_9	This state checks whether the time $T_{M\text{-}sequence}$ elapsed and the response is correct.
SM: ErrorHandling_10	In case of an incorrect response the message handler will re-send the message after a waiting time $T_{initcyc}$. After too many retries the message handler will change to the Inactive_0 state.
CheckHandler_11	Some services require several OD acquisition cycles to exchange the OD. Whenever the affected OD, ISDU, or Event handler returned to the idle state, the message handler can leave the OD acquisition loop.
Operate_12	Upon reception of a call MH_Conf_OPERATE the message handler changed to this state and after an initial time $T_{initcyc}$, it is responsible to control cyclic Process Data and On-request Data exchange according to conditions specified in Table A.9 and Table A.10. The message handler restarts on its own a new message cycle after the time t_{CYC} elapsed.
GetPD_13	The message handler used the PDTrig service to aquire PD from the Process Data handler. The message handler waits on the PD service and then changes to state GetOD_14.
GetOD_14	The message handler used the ODTrig service to aquire OD from the On-request Data handler. The message handler waits on the OD service to complement the already acquired PD and to send a message with the acquired PD/OD.
Response_15	The message handler sent a message with the acquired PD/OD.The submachine in this pseudo state waits on the response and checks its correctness.
SM: AwaitReply_16	This state checks whether the time $T_{M\text{-}sequence}$ elapsed and the response is correct.
SM: ErrorHandling_17	In case of an incorrect response the message handler will re-send the message after a waiting time $t_{CYC.}$ After too many retries the message handler will change to the Inactive_0 state.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Send a message with the requested transmission rate of COMx and with M-sequence TYPE_0: Read Direct Parameter page 1, address 0x02 ("MinCycleTime"), compiling into an M-sequence control MC = 0xA2 (see A.1.2). Start timer with $T_{M\text{-}sequence.}$
T2	1	2	Return value of "MinCycleTime" via DL_Read service confirmation.
T3	1	0	Reset timer ($T_{M\text{-}sequence}$).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T4	1	0	Reset timer ($T_{M\text{-}sequence}$).
T5	2	3	Send message using the established transmission rate, the page communication channel, and the read access option (see A.1.2). Start timer with $T_{M\text{-}sequence}$.
T6	2	3	Send message using the established transmission rate, the page communication channel, and the write access option (see A.1.2). Start timer with $T_{M\text{-}sequence}$.
T7	4	5	Reset timer ($T_{M\text{-}sequence}$).
T8	4	5	Reset timer ($T_{M\text{-}sequence}$).
T9	5	4	Re-send message after a time $T_{initcyc}$. Restart timer with $T_{M\text{-}sequence.}$
T10	3	2	Return DL_Read or DL_Write service confirmation respectively to system management.
T11	3	0	Message handler returns MH_Info (COMLOST) to DL-mode handler.
T12	2	6	-
T13	6	7	The Message handler invokes the ODTrig service for the On-request handler (see Figure 46), which is in state "ISDU_1". In this state it causes the ISDU handler to provide the OD service in correspondence to the DL_ReadParam service (see Figure 49, Transition T13).
T14	6	7	The Message handler invokes the ODTrig service for the On-request handler (see Figure 46), which is in state "ISDU_1". In this state it causes the ISDU handler to provide the OD service in correspondence to the DL_WriteParam service (see Figure 49, Transition T13).
T15	6	7	The Message handler invokes the ODTrig service for the On-request handler (see Figure 46), which is in state "ISDU_1". In this state it causes the ISDU handler to provide the OD service in correspondence to the DL_ISDUTransort service (see Figure 49, Transition T2). The message handler may need several cycles until the ISDU handler returns to the "idle" state.
T16	6	7	The Message handler invokes the ODTrig service for the On-request handler (see Figure 46), which is in state "Event_4". In this state it causes the Event handler to provide the OD service in correspondence to the EventFlag service (see Figure 53, Transition T2). The message handler may need several cycles until the Event handler returns to the "idle" state.
T17	6	7	The Message handler invokes the ODTrig service for the On-request handler (see Figure 46), which is in state "ISDU_1". In this state it causes the ISDU handler to provide the OD service in correspondence to the DL_Write service (see Figure 49, Transition T13).
T18	7	8	Send message after a recovery time $T_{initcyc}$ caused by the OD.req service. Start timer with $T_{M\text{-}sequence}$.
T19	9	10	Reset timer ($T_{M\text{-}sequence}$).
T20	9	10	Reset timer ($T_{M\text{-}sequence}$).
T21	10	9	Re-send message after a time $T_{initcyc}$. Restart timer with $T_{M\text{-}sequence}$.
T22	8	0	Message handler changes to state Inactive_0 and returns MH_Info (COMLOST) to DL-mode handler.
T23	8	11	-
T24	11	7	Acquire OD through invocation of the ODTrig service to the On-request Data handler, which in turn triggers the current handler in charge via the ISDU or EventTrig call.
T25	11	6	Return result via service primitive OD.cnf
T26	6	12	Message handler changes to state Operate_12.
T27	12	13	Start the t_{CYC} -timer. Acquire PD through invocation of the PDTrig service to the Process Data handler (see Figure 44).
T28	13	14	Acquire OD through invocation of the ODTrig service to the On-request Data handler (see Figure 46).
T29	14	15	PD and OD ready through PD.req service from PD handler and OD.req

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
			service via the OD handler. Message handler sends message. Start timer with $T_{M\text{-}sequence}$.
T30	16	17	Reset timer ($T_{M\text{-}sequence}$).
T31	16	17	Reset timer ($T_{M\text{-}sequence}$).
T32	17	16	Re-send message after a time t_{CYC}. Restart timer with $T_{M\text{-}sequence}$.
T33	15	0	Message handler changes to state Inactive_0 and returns MH_Info (COMLOST) to DL-mode handler.
T34	15	12	Device response message is correct. Return PD via service PD.cnf and via call PDTrig to the PD handler (see Table 46). Return OD via service OD.cnf and via call ODTrig to the On-request Data hander, which redirects it to the ISDU (see Table 51), Command (see Table 54), or Event handler (see Table 57) in charge.
T35	12	0	Message handler changes to state Inactive_0 and returns MH_Info (COMLOST) to the DL-mode handler.
T36	6	0	Message handler changes to state Inactive_0 and returns MH_Info (COMLOST) to the DL-mode handler.
T37	6	2	-
T38	12	2	-
T39	2	12	-

INTERNAL ITEMS	TYPE	DEFINITION
Retry	Variable	Retry counter
MaxRetry	Constant	MaxRetry = 2, see Table 97
$t_{M\text{-}sequence}$	Time	See Equation (A.6)
t_{CYC}	Time	The DL_SetMode service provides this value with its parameter "M-sequenceTime". See Equation (A.7)
$t_{initcyc}$	Time	See A.2.6
MH_Conf_xxx	Call	See Table 42

7.3.3.5 State machine of the Device message handler

Figure 42 shows the state machine of the Device message handler.

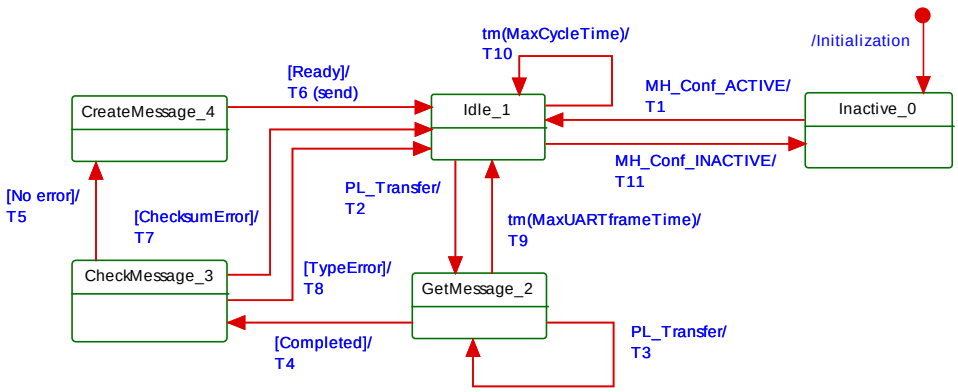


Figure 42 – State machine of the Device message handler

Table 45 shows the state transition tables of the Device message handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 45 – State transition tables of the Device message handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation by the Device DL-mode handler through MH_Conf_ACTIVE (see Table 43, Transition T1).
Idle_1	Waiting on first UART frame of the Master message through PL_Transfer service indication. Check whether time "MaxCycleTime" elapsed.
GetMessage_2	Receive a Master message UART frame. Check number of received UART frames (Device knows M-sequence type and thus knows the number of the UART frames). Check whether the time "MaxUARTframeTime" elapsed.
CheckMessage_3	Check M-sequence type and checksum of received message.
CreateMessage_4	Compile message from OD.rsp, PD.rsp, EventFlag, and PDStatus services.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	2	Start "MaxUARTframeTime" and "MaxCycleTime" when in OPERATE.
T3	2	2	Restart timer "MaxUARTframeTime".
T4	2	3	Reset timer "MaxUARTframeTime".
T5	3	4	Invoke OD.ind and PD.ind service indications
T6	4	1	Compile and invoke PL_Transfer.rsp service response (Device sends response message)
T7	3	1	Indicate error to DL-mode handler via MHInfo (CHECKSUM_MISMATCH)
T8	3	1	Indicate error to DL-mode handler via MHInfo (ILLEGAL_MESSAGETYPE)
T9	2	1	Reset both timers "MaxUARTframeTime" and "MaxCycleTime".
T10	1	1	Indicate error to DL-mode handler via MHInfo (COMLOST). Actuators shall observe this information and take corresponding actions (see 10.2 and 10.7.3).
T11	1	0	Device message handler changes state to Inactive_0.

INTERNAL ITEMS	TYPE	DEFINITION
MaxUARTFrameTime	Time	Time for the transmission of a UART frame (11 T_{BIT}) plus maximum of t_1 (1 T_{BIT}) = 11 $T_{BIT.}$
MaxCycleTime	Time	The purpose of the timer "MaxCycleTime" is to check, whether cyclic Process Data exchange took too much time or has been interrupted. MaxCycleTime shall be > MasterCycleTime (see A.3.7).
TypeError	Guard	One of the possible errors detected: ILLEGAL_MESSAGETYPE, or COMLOST
ChecksumError	Guard	Checksum error of message detected

7.3.4 Process Data handler

7.3.4.1 General

The transport of output Process Data is performed using the DL_OutputUpdate services and for input Process Data using the DL_InputTransport services (see Figure 26). A Process Data cycle is completed when the entire set of Process Data has been transferred between Master and Device in the requested direction. Such a cycle can last for more than one M-sequence.

All Process Data are transmitted within one M-sequence when using M-sequences of TYPE_2_x (see Figure 37). In this case the execution time of a Process Data cycle is equal to the cycle time t_{CYC}.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.3.4.2 Interleave mode

All Process Data and On-request Data are transmitted in this case with multiple alternating M-sequences TYPE_1_1 (Process Data) and TYPE_1_2 (On-request Data) as shown in Figure 43. It demonstrates the Master messages writing output Process Data to a Device. The service parameter PDOutAddress indicates the partition of the output PD to be transmitted (see 7.2.2.3). For input Process Data the service parameter PDInAddress correspondingly indicates the partition of the input PD. Within a Process Data cycle all input PD shall be read first followed by all output PD to be written. A Process Data cycle comprises all cycle times required to transmit the complete Process Data.

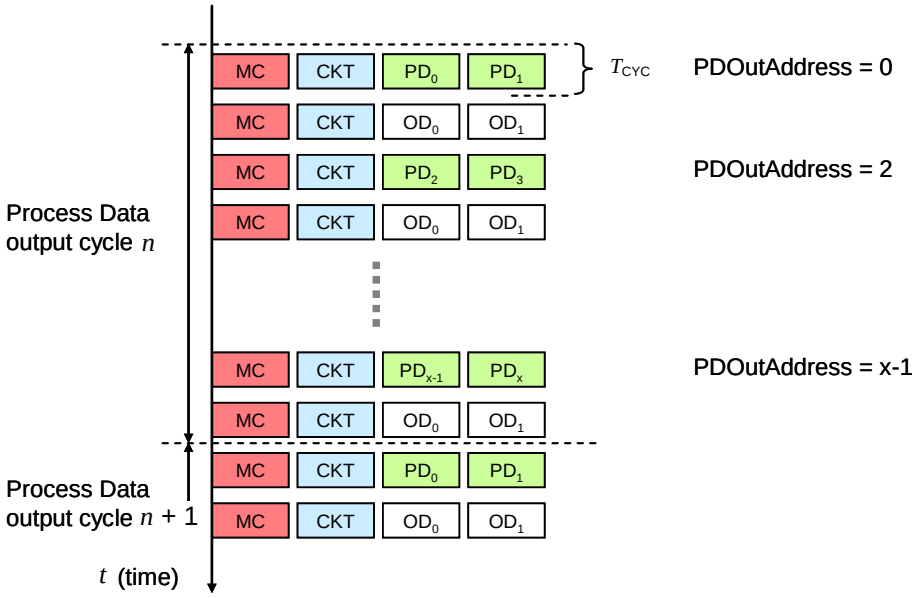


Figure 43 – Interleave mode for the segmented transmission of Process Data

Interleave mode is for legacy Devices only.

7.3.4.3 State machine of the Master Process Data handler

Figure 44 shows the state machine of the Master Process Data handler.

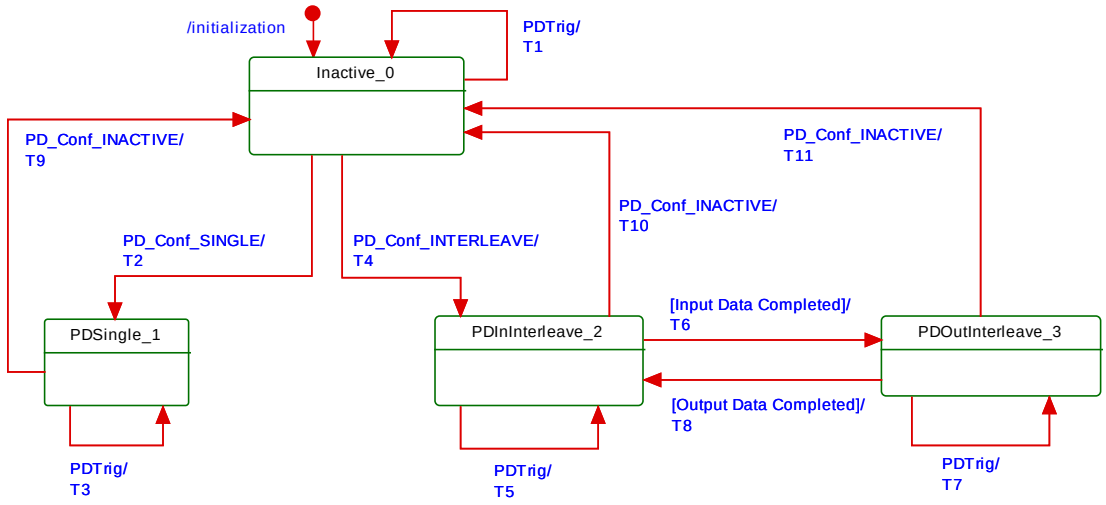


Figure 44 – State machine of the Master Process Data handler

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 46 shows the state transition tables of the Master Process Data handler.

Table 46 – State transition tables of the Master Process Data handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
PDSingle_1	Process Data communication within one single M-sequence
PDInInterleave_2	Input Process Data communication in interleave mode
PDOutInterleave_3	Output Process Data communication in interleave mode

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	0	Invoke PD.req with no Process Data
T2	0	1	NOTE The DL-mode handler configured the Process Data handler for single PD transmission (see Table 42, T10 or T11).
T3	1	1	Take data from DL_PDOutputUpdate service and invoke PD.req to propagate output PD to the message handler. Take data from PD.cnf and invoke DL_PDInputTransport.ind and DL_PDCycle.ind to propagate input PD to the AL.
T4	0	2	NOTE Configured for interleave PD transmission (see Table 42, T10 or T11).
T5	2	2	Invoke PD.req and use PD.cnf to prepare DL_PDInputTransport.ind.
T6	2	3	Invoke DL_PDInputTransport.ind and DL_PDCycle.ind to propagate input PD to the AL (see 7.2.1.11).
T7	3	3	Take data from DL_PDOutputUpdate service and invoke PD.req to propagate output PD to the message handler.
T8	3	2	Invoke DL_PDCycle.ind to indicate end of Process Data cycle to the AL (see 7.2.1.12).
T9	1	0	-
T10	2	0	-
T11	3	0	-

INTERNAL ITEMS	TYPE	DEFINITION
<None>		

7.3.4.4 State machine of the Device Process Data handler

Figure 45 shows the state machine of the Device Process Data handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

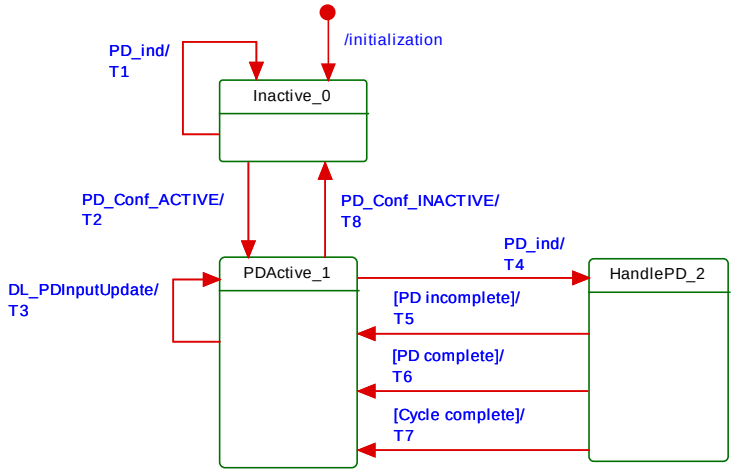


Figure 45 – State machine of the Device Process Data handler

See sequence diagrams in Figure 65 and Figure 66 for context.

Table 47 shows the state transition tables of the Device Process Data handler.

Table 47 – State transition tables of the Device Process Data handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
PDActive_1	Handler active and waiting on next message handler demand via PD service or DL_PDInputUpdate service from AL.
HandlePD_2	Check Process Data for completeness in interleave mode

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	0	Ignore Process Data
T2	0	1	-
T3	1	1	Prepare input Process Data for PD.rsp for next message handler demand
T4	1	2	Message handler demands input PD via a PD.ind service and delivers output PD or segment of output PD. Invoke PD.rsp with input Process Data when in non-interleave mode (see 7.2.2.3).
T5	2	1	-
T6	2	1	Invoke DL_PDOutputTransport.ind (see 7.2.1.9)
T7	2	1	Invoke DL_PDCycle.ind (see 7.2.1.12)
T8	1	0	-

INTERNAL ITEMS	TYPE	DEFINITION
PD_ind	Label	Invocation of service PD.ind occurred from message handler

7.3.5 On-request Data handler

7.3.5.1 General

The Master On-request Data handler is a subordinate state machine active in the "Startup_2", "PreOperate_3", and "Operate_4" state of the DL-mode handler (see Figure 33). It controls three other state machines, the so-called ISDU handler, the command handler, and the Event handler. It always starts with the ISDU handler by default.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Whenever an EventFlag.ind is received, the state machine will change to the Event handler. After the complete readout of the Event information it will return to the ISDU handler state.

Whenever a DL_Control.req or PDInStatus.ind service is received while in the ISDU handler or in the Event handler, the state machine will change to the command handler. Once the command has been served, the state machine will return to the previously active state (ISDU or Event).

7.3.5.2 State machine of the Master On-request Data handler

Figure 46 shows the Master state machine of the On-request Data handler.

The On-request Data handler redirects the ODTrig.ind service primitive for the next message content to the currently active subsidiary handler (ISDU, command, or Event). This is performed through one of the ISDUTrig, CommandTrig, or EventTrig calls.

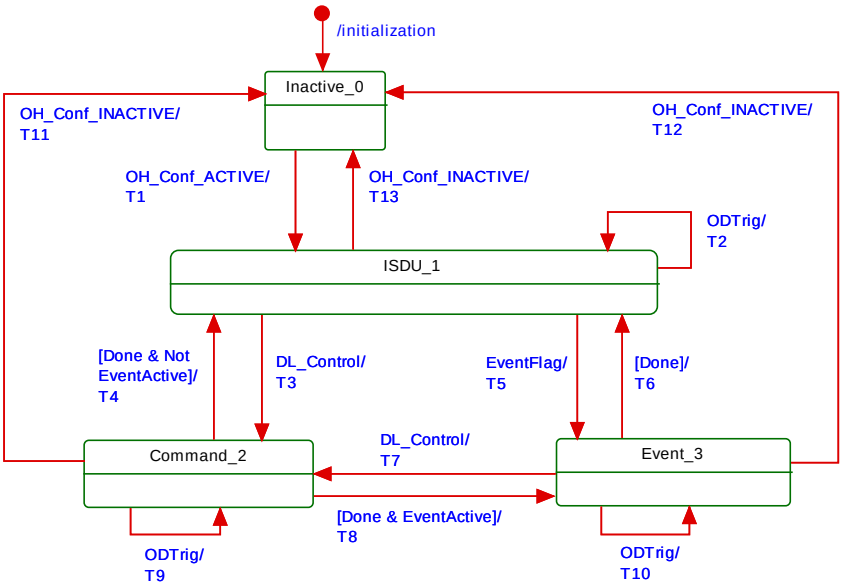


Figure 46 – State machine of the Master On-request Data handler

Table 48 shows the state transition tables of the Master On-request Data handler.

Table 48 – State transition tables of the Master On-request Data handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
ISDU_1	Default state of the on-request handler (lowest priority)
Command_2	State to control the Device via commands with highest priority
Event_3	State to convey Event information (errors, warnings, notifications) with higher priority

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	1	On-request Data handler propagates the ODTrig.ind service now named ISDUTrig to the ISDU handler (see Figure 49). In case of DL_Read, DL_Write, DL_ReadParam, or DL_WriteParam services, the ISDU handler will use a separate transition (see Figure 49, T13).
T3	1	2	-

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T4	2	1	-
T5	1	3	EventActive = TRUE
T6	3	1	EventActive = FALSE
T7	3	2	-
T8	2	3	-
T9	2	2	On-request Data handler propagates the ODTrig.ind service now named CommandTrig to the command handler (see Figure 51)
T10	3	3	On-request Data handler propagates the ODTrig.ind service now named EventTrig to the Event handler (see Figure 53)
T11	2	0	-
T12	3	0	-
T13	1	0	-

INTERNAL ITEMS	TYPE	DEFINITION
EventActive	Bool	Flag to indicate return direction after interruption of Event processing by a high priority command request

7.3.5.3 State machine of the Device On-request Data handler

Figure 47 shows the state machine of the Device On-request Data handler.

The Device On-request Data handler obtains information on the communication channel and the parameter or FlowCTRL address via the OD.ind service. The communication channels are totally independent. In case of a valid access, the corresponding ISDU, command or Event state machine is addressed via the associated communication channel.

The Device shall respond to read requests to not implemented address ranges with the value "0". It shall ignore write requests to not implemented address ranges.

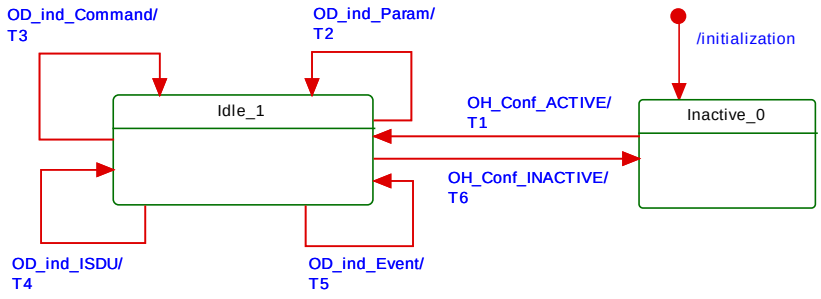


Figure 47 – State machine of the Device On-request Data handler

In case of an ISDU access in a Device without ISDU support, the Device shall respond with "No Service" (see Table A.12). An error message is not created.

NOTE OD.ind (R, ISDU, FlowCTRL = IDLE) is the default message if there are no On-request Data pending for transmission.

Table 49 shows the state transition tables of the Device On-request Data handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 49 – State transition tables of the Device On-request Data handler

STATE NAME		STATE DESCRIPTION	
Inactive_0		Waiting on activation	
Idle_1		Waiting on messages with On-request Data via service OD indication. Decomposition and analysis.	
TRANSITION	**SOURCE STATE**	**TARGET STATE**	**ACTION**
T1	0	1	-
T2	1	1	Redirect to ISDU handler (Direct Parameter page).
T3	1	1	Redirect to command handler
T4	1	1	Redirect to ISDU handler
T5	1	1	Redirect to Event handler
T6	1	0	-
INTERNAL ITEMS		**TYPE**	**DEFINITION**
OD_ind_Param		Service	Alias for Service OD.ind (R/W, PAGE, 16 to 31, Data) in case of DL_ReadParam or DL_WriteParam
OD_ind_Command		Service	Alias for Service OD.ind (W, PAGE, 0, MasterCommand)
OD_ind_ISDU		Service	Alias for Service OD.ind (R/W, ISDU, FlowCtrl, Data)
OD_ind_Event		Service	Alias for Service OD.ind (R/W, DIAGNOSIS, n, Data)

7.3.6 ISDU handler

7.3.6.1 Indexed Service Data Unit (ISDU)

The general structure of an ISDU is demonstrated in Figure 48 and specified in detail in Clause A.5.

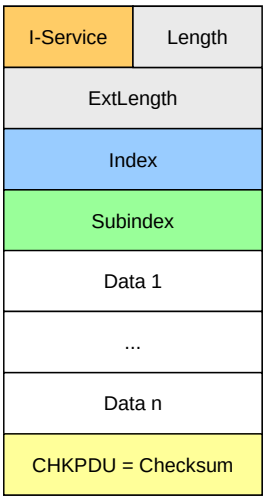


Figure 48 – Structure of the ISDU

The sequence of the elements corresponds to the transmission sequence. The elements of an ISDU can take various forms depending on the type of I-Service (see A.5.2 and Table A.12).

The ISDU allows accessing data objects (parameters and commands) to be transmitted (see Figure 5). The data objects shall be addressed by the “Index” element.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

All multi-octet data types shall be transmitted as a big-endian sequence, i.e. the most significant octet (MSO) shall be sent first, followed by less significant octets in descending order, with the least significant octet (LSO) being sent last, as shown in Figure 2.

7.3.6.2 Transmission of ISDUs

An ISDU is transmitted via the ISDU communication channel (see Figure 7 and A.1.2). A number of messages are typically required to perform this transmission (segmentation). The Master transfers an ISDU by sending an I-Service (Read/Write) request to the Device via the ISDU communication channel. It then receives the Device's response via the same channel.

In the ISDU communication channel, the "Address" element within the M-sequence control octet accommodates a counter (= FlowCTRL). FlowCTRL is controlling the segmented data flow (see A.1.2) by counting the elements of the ISDU (modulo 16) during transmission.

The Master uses the "Length" element of the ISDU and FlowCTRL to check the accomplishment of the complete transmission.

Permissible values for FlowCTRL are specified in Table 50.

Table 50 – FlowCTRL definitions

FlowCTRL	Definition
0x00 to 0x0F	COUNT Element counter within an ISDU. Increments beginning with 1 after an ISDU START. Jumps back from 15 to 0 in the Event of an overflow.
0x10	START Start of an ISDU I-Service, i.e., start of a request or a response. For the start of a request, any previously incomplete services may be rejected. For a start request associated with a response, a Device shall send “No Service” until its application returns response data (see Table A.12).
0x11	IDLE 1 No request for ISDU transmission.
0x12	IDLE 2: Reserved for future use No request for ISDU transmission.
0x13 to 0x1E	Reserved
0x1F	ABORT Abort entire service. The Master responds by rejecting received response data. The Device responds by rejecting received request data and may generate an abort.

In state Idle_1, values 0x12 to 0x1F shall not lead to a communication error.

7.3.6.3 State machine of the Master ISDU handler

Figure 49 shows the state machine of the Master ISDU handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

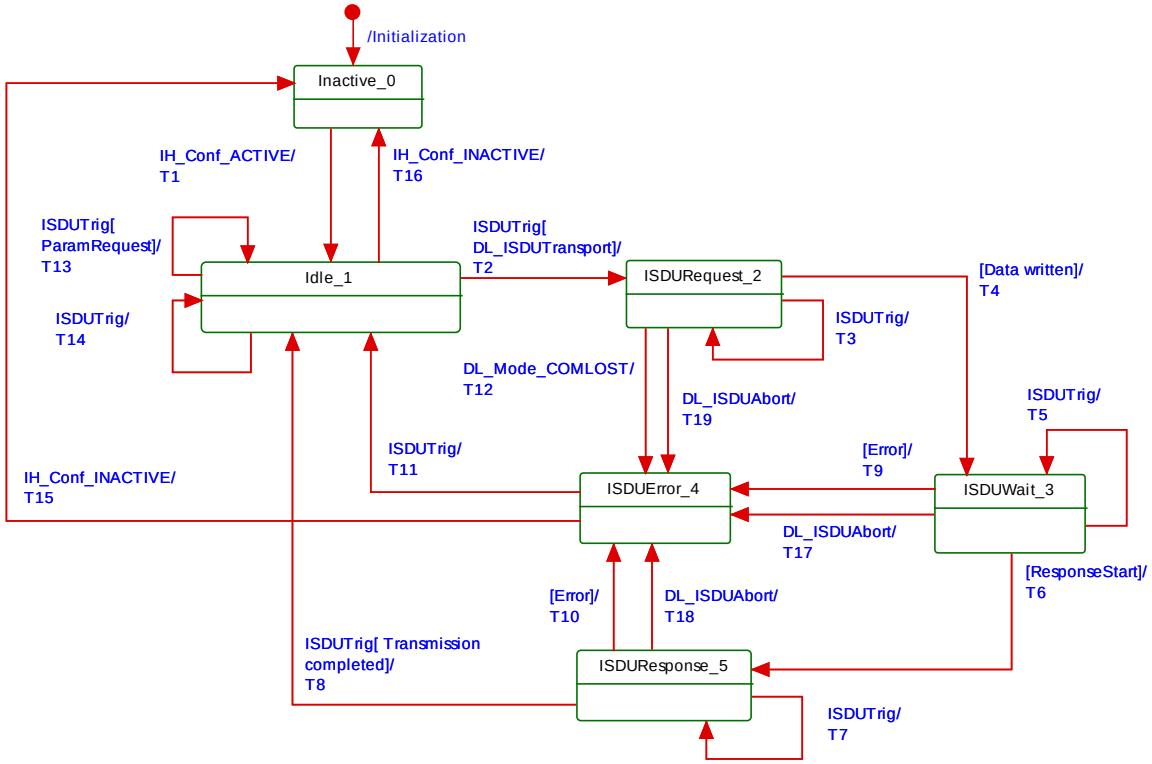


Figure 49 – State machine of the Master ISDU handler

Table 51 shows the state transition tables of the Master ISDU handler.

Table 51 – State transition tables of the Master ISDU handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
Idle_1	Waiting on transmission of next On-request Data
ISDURequest_2	Transmission of ISDU request data
ISDUWait_3	Waiting on response from Device. Observe ISDUTime
ISDUError_4	Error handling after detected errors: Invoke negative DL_ISDU_Transport response with ISDUTransportErrorInfo
ISDUResponse_5	Get response data from Device

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	2	Invoke OD.req with ISDU write start condition: OD.req (W, ISDU, flowCtrl = START, data)
T3	2	2	Invoke OD.req with ISDU data write and FlowCTRL under conditions of Table 50
T4	2	3	Start timer (ISDUTime)
T5	3	3	Invoke OD.req with ISDU read start condition: OD.req (R, ISDU, flowCtrl = START)
T6	3	5	Stop timer (ISDUTime)
T7	5	5	Invoke OD.req with ISDU data read and FlowCTRL under conditions of Table 50
T8	5	1	Invoke positive DL_ ISDUTransport confirmation

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T9	3	4	-
T10	5	4	-
T11	4	1	Invoke OD.req with ISDU abortion: OD.req (R, ISDU, flowCtrl = ABORT). Invoke negative DL_ ISDUTransport confirmation
T12	2	4	-
T13	1	1	Invoke OD.req with appropriate data. Invoke positive DL_ReadParam/DL_WriteParam confirmation
T14	1	1	Invoke OD.req with idle message: OD.req (R, ISDU, flowCtrl = IDLE)
T15	4	1	In case of lost communication the message handler informs the DL_Mode handler which in turn uses the administrative call IH_Conf_INACTIVE. No actions during this transition required.
T16	1	0	-
T17	3	4	-
T18	5	4	-
T19	2	4	-

INTERNAL ITEMS	TYPE	DEFINITION
ISDUTime	Time	Measurement of Device response time (watchdog, see Table 97)
ResponseStart	Service	OD.cnf (data different from ISDU_BUSY)
ParamRequest	Service	DL_ReadParam or DL_WriteParam
Error	Variable	Any detectable error within the ISDU transmission or DL_ISDUAbort requests, or any violation of the ISDU acknowledgement time (see Table 97)

7.3.6.4 State machine of the Device ISDU handler

Figure 50 shows the state machine of the Device ISDU handler.

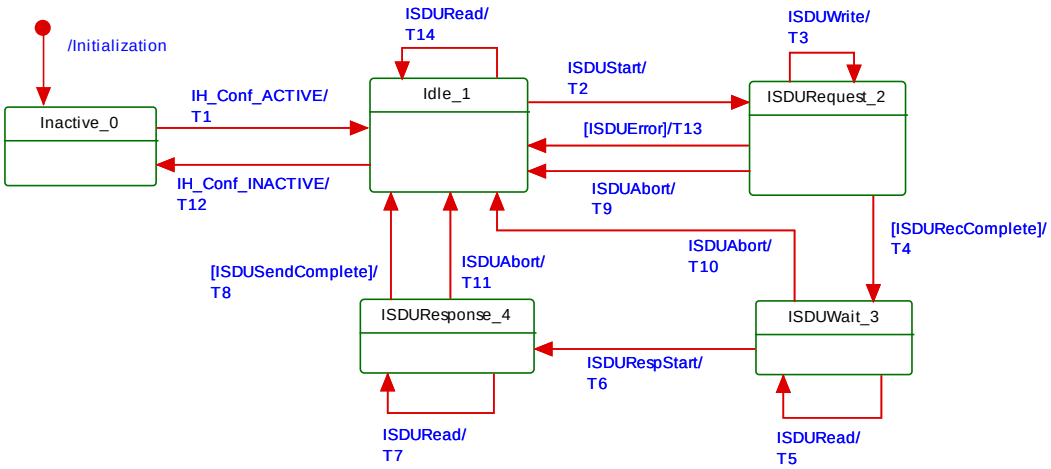


Figure 50 – State machine of the Device ISDU handler

Table 52 shows the state transition tables of the Device ISDU handler.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 52 – State transition tables of the Device ISDU handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
Idle_1	Waiting on next ISDU transmission
ISDURequest_2	Reception of ISDU request
ISDUWait_3	Waiting on data from application layer to transmit (see DL_ISDUTransport)
ISDUResponse_4	Transmission of ISDU response data

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	2	Start receiving of ISDU request data
T3	2	2	Receive ISDU request data
T4	2	3	Invoke DL_ ISDUTransport.ind to AL (see 7.2.1.6)
T5	3	3	Invoke OD.rsp with "busy" indication (see Table A.14)
T6	3	4	-
T7	4	4	Invoke OD.rsp with ISDU response data
T8	4	1	-
T9	2	1	-
T10	3	1	Invoke DL_ ISDUAbort
T11	4	1	Invoke DL_ ISDUAbort
T12	1	0	-
T13	2	1	-
T14	1	1	Invoke OD.rsp with "no service" indication (see Table A.12 and Table A.14)

INTERNAL ITEMS	TYPE	DEFINITION
ISDUStart	Service	OD.ind(W, ISDU, Start, Data)
ISDUWrite	Service	OD.ind(W, ISDU, FlowCtrl, Data)
ISDURecComplete	Guard	If OD.ind(R, ISDU, Start, ...) received
ISDURespStart	Service	DL_ ISDUTransport.rsp()
ISDURead	Service	OD.ind(R, ISDU, Start or FlowCtrl, ...)
ISDUSendComplete	Guard	If OD.ind(R, ISDU, IDLE, ...) received
ISDUAbort	Service	OD.ind(R/W, ISDU, Abort, ...)
ISDUError	Guard	If ISDU structure is incorrect

7.3.7 Command handler

7.3.7.1 General

The command handler passes the control code (PDOUTVALID or PDOUTINVALID) contained in the DL_Control.req service primitive to the cyclically operating message handler via the OD.req service and MasterCommands. The message handler uses the page communication channel.

The permissible control codes for output Process Data are listed in Table 53.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 53 – Control codes

Control code	MasterCommand	Description
PDOUTVALID	ProcessDataOutputOperate	Output Process Data valid
PDOUTINVALID	DeviceOperate	Output Process Data invalid or missing

The command handler receives input Process Data status information via the PDInStatus service and propagates it within a DL_Control.ind service primitive.

In addition, the command handler translates Device mode change requests from system management into corresponding MasterCommands (see Table B.2).

7.3.7.2 State machine of the Master command handler

Figure 51 shows the state machine of the Master command handler.

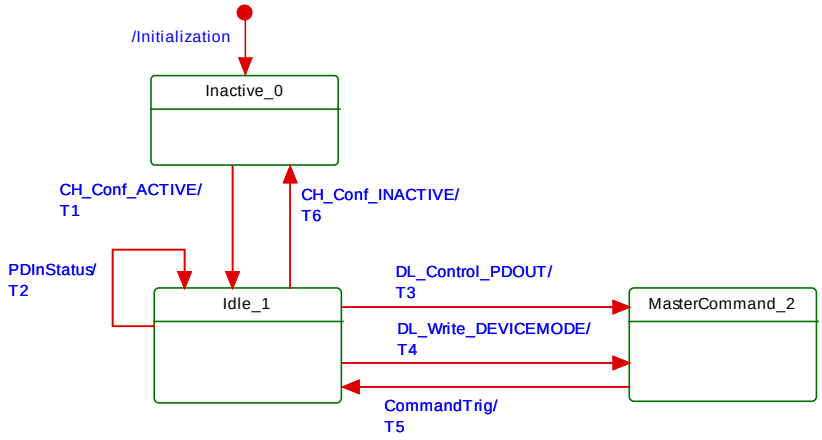


Figure 51 – State machine of the Master command handler

Table 54 shows the state transition tables of the Master command handler.

Table 54 – State transition tables of the Master command handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation by DL-mode handler
Idle_1	Waiting on new command from AL: DL_Control (status of output PD) or from SM: DL_Write (change Device mode, for example to OPERATE), or waiting on PDInStatus.ind service primitive.
MasterCommand_2	Prepare data for OD.req service primitive. Waiting on demand from OD handler (CommandTrig).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	1	If service PDInStatus.ind = VALID invoke DL_Control.ind (VALID) to signal valid input Process Data to AL. If service PDInStatus.ind = INVALID invoke DL_Control.ind (INVALID) to signal invalid input Process Data to AL.
T3	1	1	If service DL_Control.req = PDOUTVALID invoke OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x98). If service DL_Control.req = PDOUTINVALID invoke OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x99). See Table B.2.
T4	1	2	The services DL_Write_DEVICEMODE translate into:

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
			INACTIVE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x5A) STARTUP: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x97) PREOPERATE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x9A) OPERATE: OD.req (WRITE, PAGE, 0, 1, MasterCommand = 0x99)
T5	2	1	A call CommandTrig from the OD handler causes the command handler to invoke the OD.req service primitive and subsequently the message handler to send the appropriate MasterCommand to the Device.
T6	1	0	-

INTERNAL ITEMS	TYPE	DEFINITION
DEVICEMODE	Label	Any of the Device modes: INACTIVE, STARTUP, PREOPERATE, or OPERATE
PDOUT	Label	Any of the two output control codes: PDOUTVALID or PDOUTINVALID (see Table 53)

7.3.7.3 State machine of the Device command handler

Figure 52 shows the Device state machine of the command handler. It is mainly driven by MasterCommands from the Master's command handler to control the Device modes and the status of output Process Data. It also controls the status of input Process Data via the PDInStatus service.

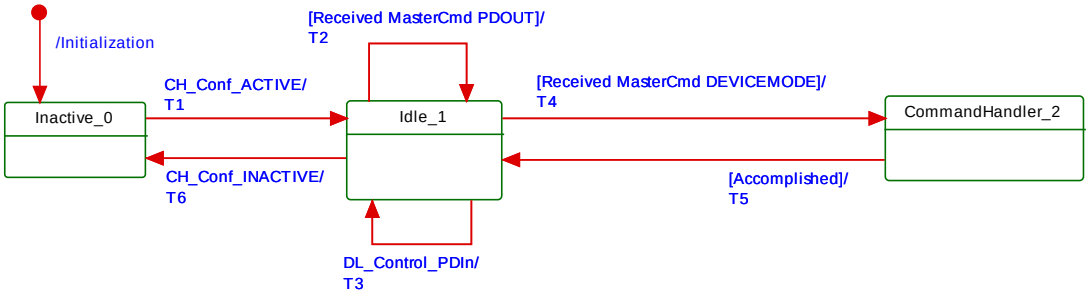


Figure 52 – State machine of the Device command handler

Table 55 shows the state transition tables of the Device command handler.

Table 55 – State transition tables of the Device command handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
Idle_1	Waiting on next MasterCommand
CommandHandler_2	Decompose MasterCommand and invoke specific actions (see B.1.2): If MasterCommand = 0x5A then change Device state to INACTIVE. If MasterCommand = 0x97 then change Device state to STARTUP. If MasterCommand = 0x9A then change Device state to PREOPERATE. If MasterCommand = 0x99 then change Device state to OPERATE.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	1	Invoke DL_Control.ind (PDOUTVALID) if received MasterCommand = 0x98. Invoke DL_Control.ind (PDOUTINVALID) if received MasterCommand = 0x99.
T3	1	1	If service DL_Control.req (VALID) then invoke PDInStatus.req (VALID). If service DL_Control.req (INVALID) then invoke PDInStatus.req

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
			(INVALID). Message handler uses PDInStatus service to set/reset the PD status flag (see A.1.5)
T4	1	2	-
T5	2	1	-
T6	1	0	-

INTERNAL ITEMS	TYPE	DEFINITION
<none>		

7.3.8 Event handler

7.3.8.1 Events

There are two types of Events, one without details, and another one with details. Events without details may have been implemented in legacy Devices, but they shall not be used for Devices in accordance with this standard. However, all Masters shall support processing of both Events with details and Events without details.

The general structure and coding of Events is specified in A.6. Event codes without details are specified in Table A.16. EventCodes with details are specified in Annex D. The structure of the Event memory for EventCodes with details within a Device is specified in Table 56.

Table 56 – Event memory

Address	Event slot number	Parameter Name	Description
0x00		StatusCode	Summary of status and error information. Also used to control read access for individual messages.
0x01	1	EventQualifier 1	Type, mode and source of the Event
0x02		EventCode 1	16-bit EventCode of the Event
0x03			
0x04	2	EventQualifier 2	Type, mode and source of the Event
0x05		EventCode 2	16-bit EventCode of the Event
0x06			
...			
0x10	6	EventQualifier 6	Type, mode and source of the Event
0x11		EventCode 6	16-bit EventCode of the Event
0x12			
0x13 to 0x1F			Reserved for future use

7.3.8.2 Event processing

The Device AL writes an Event to the Event memory and then sets the "Event flag" bit, which is sent to the Master in the next message within the CKS octet (see 7.3.3.2 and A.1.5).

Upon reception of a Device reply message with the "Event flag" bit = 1, the Master shall switch from the ISDU handler to the Event handler. The Event handler starts reading the StatusCode.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

If the "Event Details" bit is set (see Figure A.23), the Master shall read the Event details of the Events indicated in the StatusCode from the Event memory. Once it has read an Event detail, it shall invoke the service DL_Event.ind. After reception of the service DL_EventConf, the Master shall write any data to the StatusCode to reset the "Event flag" bit. The Event handling on the Master shall be completed regardless of the contents of the Event data received (EventQualifier, EventCode).

If the "Event Details" bit is not set (see Figure A.22) the Master Event handler shall generate the standardized Events according to Table A.16 beginning with the most significant bit in the EventCode.

Write access to the StatusCode indicates the end of Event processing to the Device. The Device shall ignore the data of this Master Write access. The Device then resets the "Event flag" bit and may now change the content of the fields in the Event memory.

7.3.8.3 State machine of the Master Event handler

Figure 53 shows the Master state machine of the Event handler.

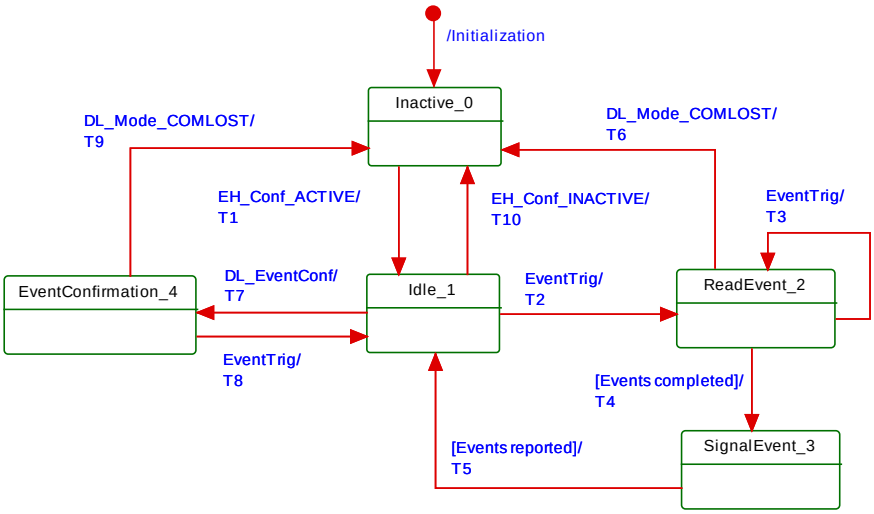


Figure 53 – State machine of the Master Event handler

Table 57 shows the state transition tables of the Master Event handler.

Table 57 – State transition tables of the Master Event handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
Idle_1	Waiting on next Event indication ("EventTrig" through On-request Data handler) or Event confirmation through service DL_EventConf from Master AL.
ReadEvent_2	Read Event data set from Device message by message through Event memory address. Check StatusCode for number of activated Events (see Table 56).
SignalEvent_3	Analyze Event data and invoke DL_Event indication to Master AL (see 7.2.1.15) for each available Event.
EventConfirmation_4	Waiting on Event confirmation transmission via service OD.req to the Device

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	2	Read Event StatusCode octet via service OD.req (R, DIAGNOSIS, Event

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
			memory address = 0, 1)
T3	2	2	Read octets from Event memory via service OD.req (R, DIAGNOSIS, incremented Event memory address, 1)
T4	2	3	-
T5	3	1	-
T6	2	0	-
T7	1	4	-
T8	4	1	Invoke OD.req (W, DIAGNOSIS, 0, 1, any data) with Write access to "StatusCode" (see Table 56) to confirm Event readout to Device
T9	4	0	-
T10	1	0	-

INTERNAL ITEMS	TYPE	DEFINITION
<None>		

7.3.8.4 State machine of the Device Event handler

Figure 54 shows the state machine of the Device Event handler.

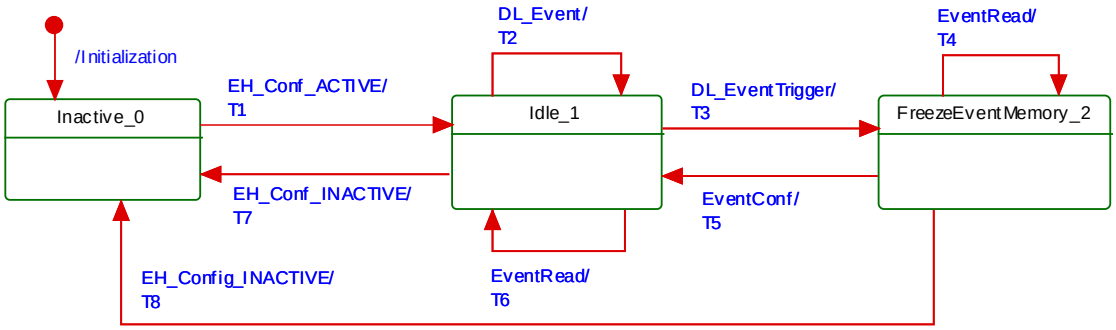


Figure 54 – State machine of the Device Event handler

Table 58 shows the state transition tables of the Device Event handler.

Table 58 – State transition tables of the Device Event handler

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on activation
Idle_1	Waiting on DL-Event service from AL providing Event data and the DL_EventTrigger service to fire the "Event flag" bit (see A.1.5)
FreezeEventMemory_2	Waiting on readout of the Event memory and on Event memory readout confirmation through write access to the StatusCode

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	1	Change Event memory entries with new Event data (see Table 56)
T3	1	2	Invoke service EventFlag.req (Flag = TRUE) to indicate Event activation to the Master via the "Event flag" bit. Mark all Event slots in memory as not changeable.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T4	2	2	Master requests Event memory data via EventRead (= OD.ind). Send Event data by invoking OD.rsp with Event data of the requested Event memory address.
T5	2	1	Invoke service EventFlag.req (Flag = FALSE) to indicate Event deactivation to the Master via the "Event flag" bit. Mark all Event slots in memory as invalid according to A.6.3.
T6	1	1	Send contents of Event memory by invoking OD.rsp with Event data
T7	1	0	-
T8	2	0	Discard Event memory data

INTERNAL ITEMS	TYPE	DEFINITION
EventRead	Service	OD.ind (R, DIAGNOSIS, Event memory address, length, data)
EventConf	Service	OD.ind (W, DIAGNOSIS, address = 0, data = don't care)

8 Application layer (AL)

8.1 General

Figure 55 shows an overview of the structure and services of the Master application layer (AL).

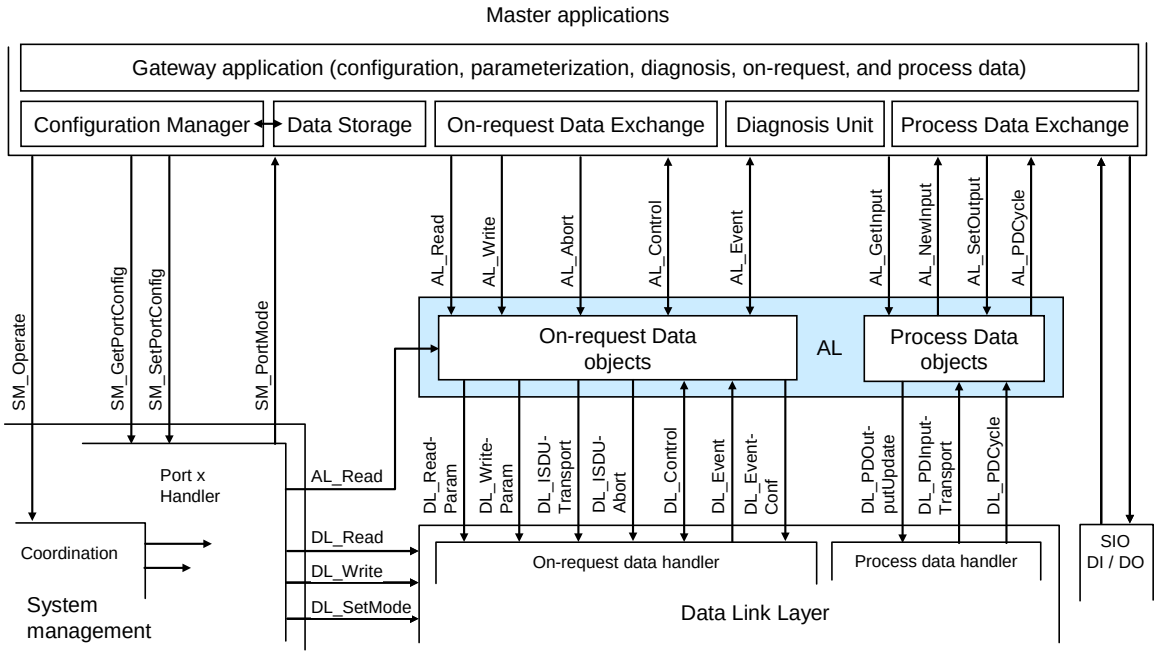


Figure 55 – Structure and services of the application layer (Master)

Figure 56 shows an overview of the structure and services of the Device application layer (AL).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

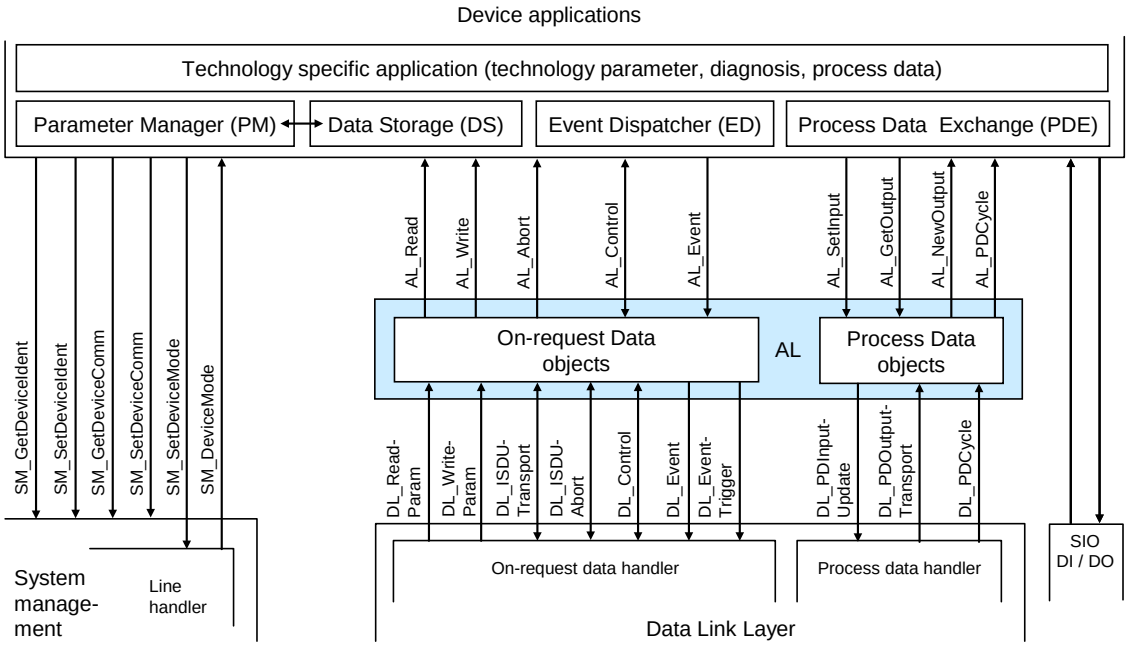


Figure 56 – Structure and services of the application layer (Device)

8.2 Application layer services

8.2.1 AL services within Master and Device

Clause 8 defines the services of the application layer (AL) to be provided to the Master and Device applications and system management via its external interfaces. Table 59 lists the assignments of Master and Device to their roles as initiator or receiver for the individual AL services. Empty fields indicate no availability of this service on Master or Device.

Table 59 – AL services within Master and Device

Service name	Master	Device
AL_Read	R	I
AL_Write	R	I
AL_Abort	R	I
AL_GetInput	R	
AL_NewInput	I	
AL_SetInput		R
AL_PDCycle	I	I
AL_GetOutput		R
AL_NewOutput		I
AL_SetOutput	R	
AL_Event	I / R	R
AL_Control	I / R	I / R
Key (see 3.3.4) I Initiator of service R Receiver (Responder) of service		

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.2.2 AL Services

8.2.2.1 AL_Read

The AL_Read service is used to read On-request Data from a Device connected to a specific port. The parameters of the service primitives are listed in Table 60.

Table 60 – AL_Read

Parameter name	.req	.ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Subindex	M	M		
Result (+)			S	S(=)
Port				M
Data			M	M(=)
Result (-)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the port number for the On-request Data to be read.

Parameter type: Unsigned8

Index
This parameter indicates the address of On-request Data objects to be read from the Device. Index 0 in conjunction with Subindex 0 addresses the entire set of Direct Parameters from 0 to 15 (see Direct Parameter page 1 in Table B.1) or in conjunction with Subindices 1 to 16 the individual parameters from 0 to 15. Index 1 in conjunction with Subindex 0 addresses the entire set of Direct Parameters from addresses 16 to 31 (see Direct Parameter page 2 in Table B.1) or in conjunction with Subindices 1 to 16 the individual parameters from 16 to 31. It uses the page communication channel (see Figure 6) for both and always returns a positive result. For all the other indices (see B.2) the ISDU communication channel is used.

Permitted values: 0 to 65 535 (See B.2.1 for constraints)

Subindex
This parameter indicates the element number within a structured On-request Data object. A value of 0 indicates the entire set of elements.

Permitted values: 0 to 255

Result (+):
This selection parameter indicates that the service has been executed successfully.

Port
This parameter contains the port number of the requested On-request Data.

Data
This parameter contains the read values of the On-request Data.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Parameter type: Octet string

Result (-):
This selection parameter indicates that the service failed.

Port
This parameter contains the port number for the requested On-request Data.

ErrorInfo
This parameter contains error information.

Permitted values: see Annex C

NOTE The AL maps DL ErrorInfos into its own AL ErrorInfos using Annex C.

8.2.2.2 AL_Write

The AL_Write service is used to write On-request Data to a Device connected to a specific port. The parameters of the service primitives are listed in Table 61.

Table 61 – AL_Write

Parameter name	.req	.ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Subindex	M	M		
Data	M	M(=)		
Result (+)			S	S(=)
Port				M
Result (-)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the port number for the On-request Data to be written.

Parameter type: Unsigned8

Index
This parameter indicates the address of On-request Data objects to be written to the Device. Index 0 always returns a negative result. Index 1 in conjunction with Subindex 0 addresses the entire set of Direct Parameters from addresses 16 to 31 (see Direct Parameter page 2 in Table B.1) or in conjunction with subindices 1 to 16 the individual parameters from 16 to 31. It uses the page communication channel (see Figure 6) in case of Index 1 and always returns a positive result. For all the other Indices (see Clause B.2) the ISDU communication channel is used.

Permitted values: 1 to 65 535 (see Table 97)

Subindex
This parameter indicates the element number within a structured On-request Data object. A value of 0 indicates the entire set of elements.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permitted values: 0 to 255

Data
This parameter contains the values of the On-request Data.

Parameter type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

Port
This parameter contains the port number of the On-request Data.

Result (-):
This selection parameter indicates that the service failed.

Port
This parameter contains the port number of the On-request Data.

ErrorInfo
This parameter contains error information.

Permitted values: see Annex C

8.2.2.3 AL_Abort

The AL_Abort service is used to abort a current AL_Read or AL_Write service on a specific port. Invocation of this service abandons the response to an AL_Read or AL_Write service in progress on the Master. The parameters of the service primitives are listed in Table 62.

Table 62 – AL_Abort

Parameter name	.req	.ind
Argument	M	M
Port	M	

Argument
The service-specific parameter is transmitted in the argument.

Port
This parameter contains the port number of the service to be abandoned.

8.2.2.4 AL_GetInput

The AL_GetInput service reads the input data within the Process Data provided by the data link layer of a Device connected to a specific port. The parameters of the service primitives are listed in Table 63.

Table 63 – AL_GetInput

Parameter name	.req	.cnf
Argument	M	
Port	M	
Result (+)		S
Port		M

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Parameter name	.req	.cnf
InputData		M
Result (-)		S
Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the port number for the Process Data to be read.

Result (+):
This selection parameter indicates that the service has been executed successfully.

Port
This parameter contains the port number for the Process Data.

InputData
This parameter contains the values of the requested process input data of the specified port.

Parameter type: Octet string

Result (-):
This selection parameter indicates that the service failed.

Port
This parameter contains the port number for the Process Data.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_DATA (DL did not provide Process Data)

8.2.2.5 AL_NewInput

The AL_NewInput local service indicates the receipt of updated input data within the Process Data of a Device connected to a specific port. The parameters of the service primitives are listed in Table 64.

Table 64 – AL_NewInput

Parameter name	.ind
Argument	M
Port	M

Argument
The service-specific parameter is transmitted in the argument.

Port
This parameter specifies the port number of the received Process Data.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.2.2.6 AL_SetInput

The AL_SetInput local service updates the input data within the Process Data of a Device. The parameters of the service primitives are listed in Table 65.

Table 65 – AL_SetInput

Parameter name	.req	.cnf
Argument	M	
InputData	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

InputData
This parameter contains the Process Data values of the input data to be transmitted.

Parameter type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (Service unavailable within current state)

8.2.2.7 AL_PDCycle

The AL_PDCycle local service indicates the end of a Process Data cycle. The Device application can use this service to transmit new input data to the application layer via AL_SetInput. The parameters of the service primitives are listed in Table 66.

Table 66 – AL_PDCycle

Parameter name	.ind
Argument	
Port	O

Argument
The service-specific parameter is transmitted in the argument.

Port
This parameter contains the port number of the received new Process Data (Master only).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.2.2.8 AL_GetOutput

The AL_GetOutput service reads the output data within the Process Data provided by the data link layer of the Device. The parameters of the service primitives are listed in Table 67.

Table 67 – AL_GetOutput

Parameter name	.req	.cnf
Argument	M	
Result (+)		S
OutputData		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Result (+):
This selection parameter indicates that the service has been executed successfully.

OutputData
This parameter contains the Process Data values of the requested output data.

Parameter type: Octet string

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
NO_DATA (DL did not provide Process Data)

8.2.2.9 AL_NewOutput

The AL_NewOutput local service indicates the receipt of updated output data within the Process Data of a Device. This service has no parameters. The service primitives are shown in Table 68.

Table 68 – AL_NewOutput

Parameter name	.ind
<None>	

8.2.2.10 AL_SetOutput

The AL_SetOutput local service updates the output data within the Process Data of a Master. The parameters of the service primitives are listed in Table 69.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 69 – AL_SetOutput

Parameter name	.req	.cnf
Argument	M	
Port	M	
OutputData	M	
Result (+)		S
Port		M
Result (-)		S
Port		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the port number of the Process Data to be written.

OutputData
This parameter contains the output data to be written at the specified port.

Parameter type: Octet string

Result (+):
This selection parameter indicates that the service has been executed successfully.

Port
This parameter contains the port number for the Process Data.

Result (-):
This selection parameter indicates that the service failed.

Port
This parameter contains the port number for the Process Data.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (Service unavailable within current state)

8.2.2.11 AL_Event

The AL_Event service indicates up to 6 pending status or error messages. The source of one Event can be local (Master) or remote (Device). The Event can be triggered by a communication layer or by an application. The parameters of the service primitives are listed in Table 70.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 70 – AL_Event

Parameter name		.req	.ind	.rsp	.cnf
Argument		M	M	M	M
Port			M	M	M
EventCount		M	M		
Event(1)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origin		M		
	EventCode	M	M		
...					
Event(n)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origin		M		
	EventCode	M	M		

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the port number of the Event data.

EventCount
This parameter indicates the number n (1 to 6) of Events in the Event memory.

Event(x)
Depending on EventCount this parameter exists n times. Each instance contains the following elements.

Instance
This parameter indicates the Event source.

Permitted values: Application (see Table A.17)

Mode
This parameter indicates the Event mode.

Permitted values: SINGLESHOT, APPEARS, DISAPPEARS (see Table A.20)

Type
This parameter indicates the Event category.

Permitted values: ERROR, WARNING, NOTIFICATION (see Table A.19)

Origin
This parameter indicates whether the Event was generated in the local communication section or remotely (in the Device).

Permitted values: LOCAL, REMOTE

EventCode
This parameter contains a code identifying a certain Event.

Permitted values: see Annex D

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.2.2.12 AL_Control

The AL_Control service contains the Process Data qualifier status information transmitted to and from the Device application. The parameters of the service primitives are listed in Table 71.

Table 71 – AL_Control

Parameter name	.req	.ind
Argument	M	M
Port	C	C
ControlCode	M	M

Argument
The service-specific parameters are transmitted in the argument.

Port
This parameter contains the number of the related port.

ControlCode
This parameter contains the qualifier status of the Process Data (PD).

Permitted values:

VALID	(Input Process Data valid)
INVALID	(Input Process Data invalid)
PDOUTVALID	(Output Process Data valid, see Table 53)
PDOUTINVALID	(Output Process Data invalid, see Table 53)

8.3 Application layer protocol

8.3.1 Overview

Figure 7 shows that the application layer offers services for data objects which are transformed into the special communication channels of the data link layer.

The application layer manages the data transfer with all its assigned ports. That means, AL service calls need to identify the particular port they are related to.

8.3.2 On-request Data transfer

8.3.2.1 OD state machine of the Master AL

Figure 57 shows the state machine for the handling of On-request Data (OD) within the application layer.

"AL_Service" represents any AL service in Table 59 related to OD. "Portx" indicates a particular port number.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

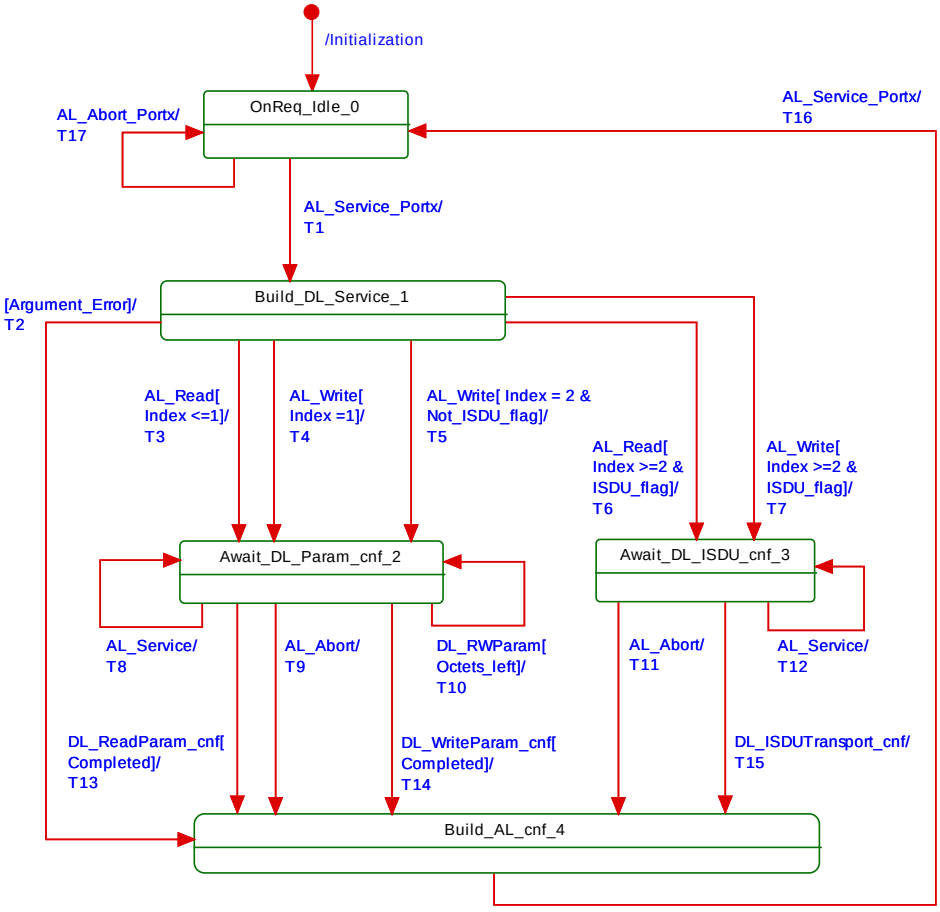


Figure 57 – OD state machine of the Master AL

Table 72 shows the states and transitions for the OD state machine of the Master AL.

Table 72 – States and transitions for the OD state machine of the Master AL

STATE NAME	STATE DESCRIPTION
OnReq_Idle_0	AL service invocations from the Master applications or from the SM Portx handler (see Figure 55) can be accepted within this state.
Build_DL_Service_1	Within this state AL service calls are checked and corresponding DL services are created within the subsequent states. In case of an error in the arguments of the AL service a negative AL confirmation is created and returned.
Await_DL_Param_cnf_2	Within this state the AL service call is transformed in a sequence of as many DL_ReadParam or DL_WriteParam calls as needed (Direct Parameter page access; see page communication channel in Figure 6). All asynchronously occurred AL service invocations except AL_Abort are rejected (see 3.3.7).
Await_DL_ISDU_cnf_3	Within this state the AL service call is transformed in a DL_ISDUTransport service call (see ISDU communication channel in Figure 6). All asynchronously occurred AL service invocations except AL_Abort are rejected (see 3.3.7).
Build_AL_cnf_4	Within this state an AL service confirmation is created depending on an argument error, the DL service confirmation, or an AL_Abort.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Memorize the port number "Portx".
T2	1	4	Prepare negative AL service confirmation.
T3	1	2	Prepare DL_ReadParam for Index 0 or 1.
T4	1	2	Prepare DL_WriteParam for Index 1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T5	1	2	Prepare DL_WriteParam for Index 2 if the Device does not support ISDU.
T6	1	3	Prepare DL_ ISDUTransport(read)
T7	1	3	Prepare DL_ ISDUTransport(write)
T8	2	2	Return negative AL service confirmation on this asynchronous service call.
T9	2	4	All current DL service actions are abandoned and a negative AL service confirmation is prepared.
T10	2	2	Call next DL_ReadParam or DL_WriteParam service if not all OD are transferred.
T11	3	4	All current DL service actions are abandoned and a negative AL service confirmation is prepared.
T12	3	3	Return negative AL service confirmation on this asynchronous service call.
T13	2	4	Prepare positive AL service confirmation.
T14	2	4	Prepare positive AL service confirmation.
T15	3	4	Prepare positive AL service confirmation.
T16	4	0	Return positive AL service confirmation with port number "Portx".
T17	0	0	Return negative AL service confirmation with port number "Portx".

INTERNAL ITEMS	TYPE	DEFINITION
Argument_Error	Bool	Illegal values within the service body, for example "Port number or Index out of range"
DL_RWParam	Label	"DL_RWParam": DL_WriteParam_cnf or DL_ReadParam_cnf
Completed	Bool	No more OD left for transfer
Octets_left	Bool	More OD for transfer
Portx	Variable	Service body variable indicating the port number
ISDU_Flag	Bool	Device supports ISDU
AL_Service	Label	"AL_Service" represents any AL service in Table 59 related to OD

8.3.2.2 OD state machine of the Device AL

Figure 58 shows the state machine for the handling of On-request Data (OD) within the application layer of a Device.

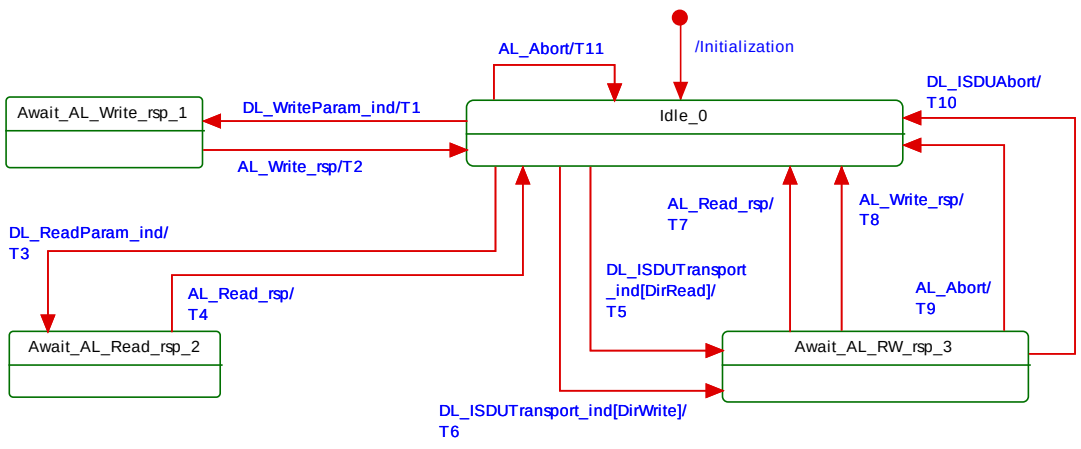


Figure 58 – OD state machine of the Device AL

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 73 shows the states and transitions for the OD state machine of the Device AL.

Table 73 – States and transitions for the OD state machine of the Device AL

STATE NAME	STATE DESCRIPTION
Idle_0	The Device AL is waiting on subordinated DL service calls triggered by Master messages.
Await_AL_Write_rsp_1	The Device AL is waiting on a response from the technology specific application (write access to Direct Parameter page).
Await_AL_Read_rsp_2	The Device AL is waiting on a response from the technology specific application (read access to Direct Parameter page).
Await_AL_RW_rsp_3	The Device AL is waiting on a response from the technology specific application (read or write access via ISDU).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Invoke AL_Write.
T2	1	0	Invoke DL_WriteParam (16 to 31).
T3	0	2	Invoke AL_Read.
T4	2	0	Invoke DL_ReadParam (0 to 31).
T5	0	3	Invoke AL_Read.
T6	0	3	Invoke AL_Write.
T7	3	0	Invoke DL_ ISDUTransport(read)
T8	3	0	Invoke DL_ ISDUTransport(write)
T9	3	0	Current AL_Read or AL_Write abandoned upon this asynchronous AL_Abort service call. Return negative DL_ ISDUTransport (see 3.3.7).
T10	3	0	Current waiting on AL_Read or AL_Write abandoned.
T11	0	0	Current DL_ ISDUTransport abandoned. All OD are set to "0".

INTERNAL ITEMS	TYPE	DEFINITION
DirRead	Bool	Access direction: DL_ ISDUTransport(read) causes an AL_Read
DirWrite	Bool	Access direction: DL_ ISDUTransport(write) causes an AL_Read

8.3.2.3 Sequence diagrams for On-request Data

Figure 59 through Figure 61 demonstrate complete interactions between Master and Device for several On-request Data exchange use cases.

Figure 59 demonstrates two examples for the exchange of On-request Data. For Indices > 1 this is performed with the help of ISDUs and corresponding DL services (ISDU communication channel according to Figure 6). Access to Direct Parameter pages 0 and 1 uses different DL services (page communication channel according to Figure 6).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

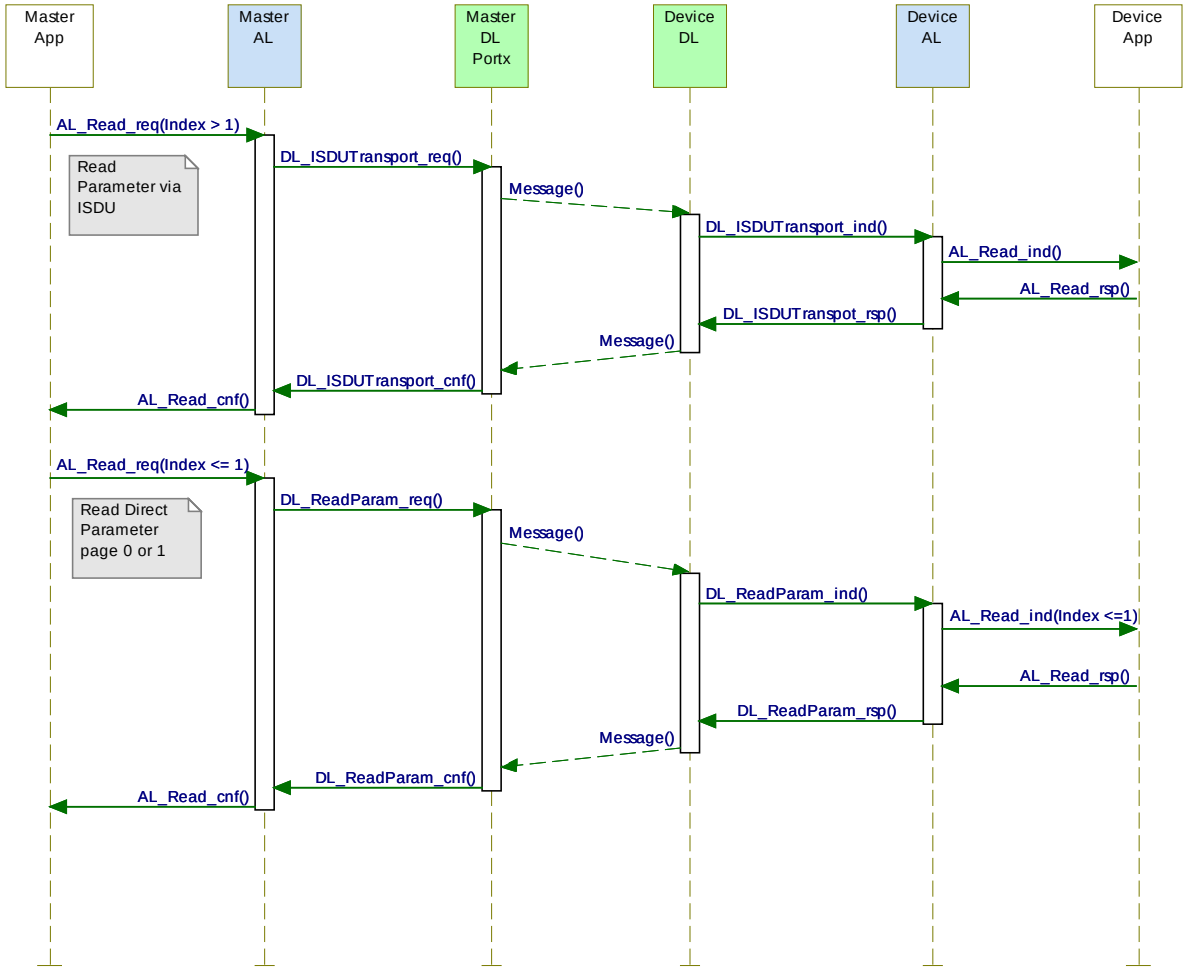


Figure 59 – Sequence diagram for the transmission of On-request Data

Figure 60 demonstrates the behaviour of On-request Data exchange in case of an error such as requested Index not available (see Table C.1).

Another possible error occurs when the Master application (gateway) tries to read an Index > 1 from a Device, which does not support ISDU. The Master AL would respond immediately with "NO_ISDU_SUPPORTED" as the features of the Device are acquired during start-up through reading the Direct Parameter page 1 via the parameter "M-sequence Capability" (see Table B.1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

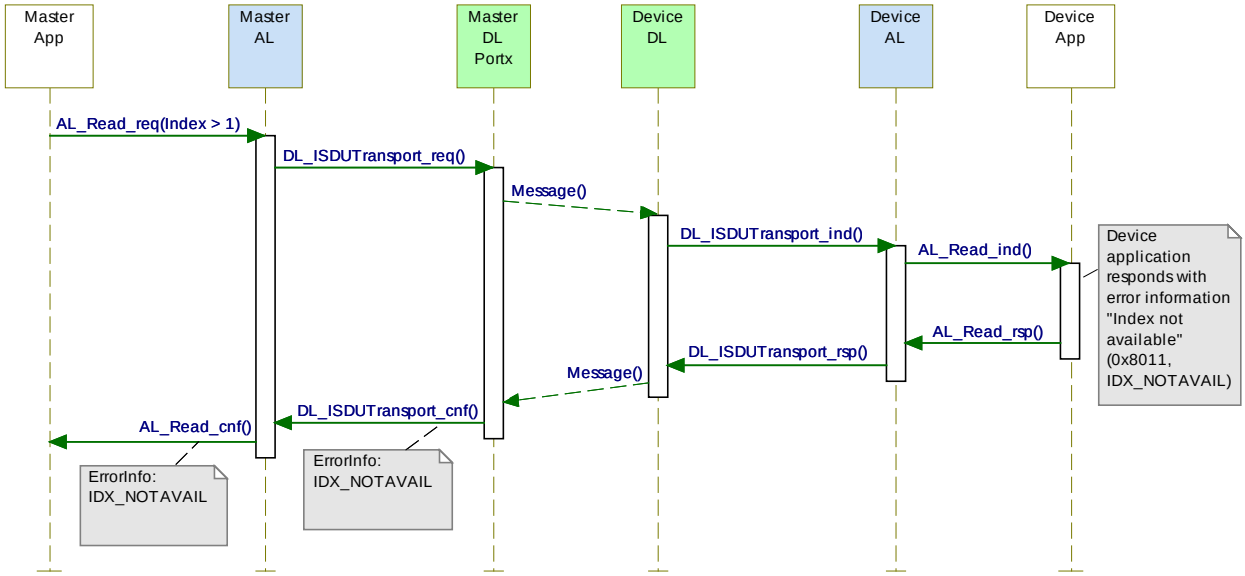


Figure 60 – Sequence diagram for On-request Data in case of errors

Figure 61 demonstrates the behaviour of On-request Data exchange in case of an ISDU timeout (5 500 ms). A Device shall respond within less than the "ISDU acknowledgement time" (see 10.7.5).

NOTE See Table 97 for system constants such as "ISDU acknowledgement time".

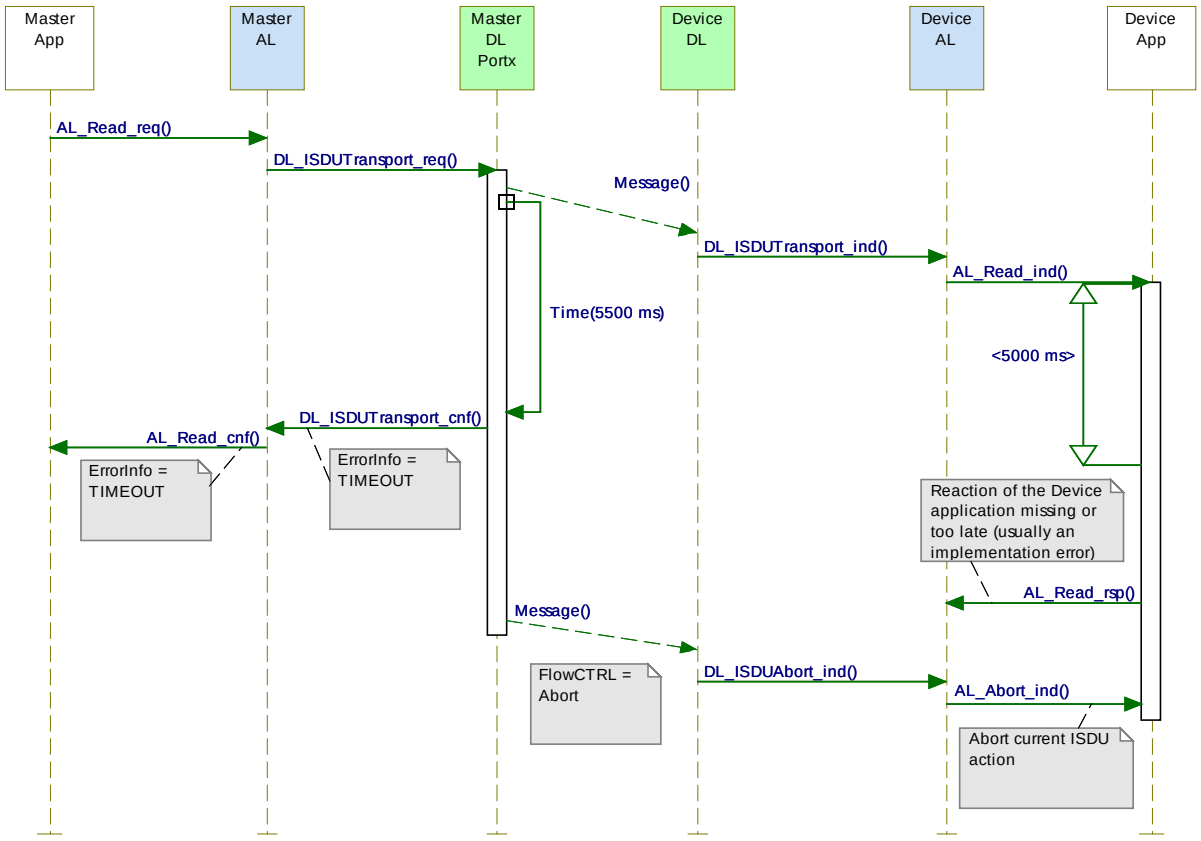


Figure 61 – Sequence diagram for On-request Data in case of timeout

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.3.3 Event processing

8.3.3.1 Event state machine of the Master AL

Figure 62 shows the Event state machine of the Master application layer.

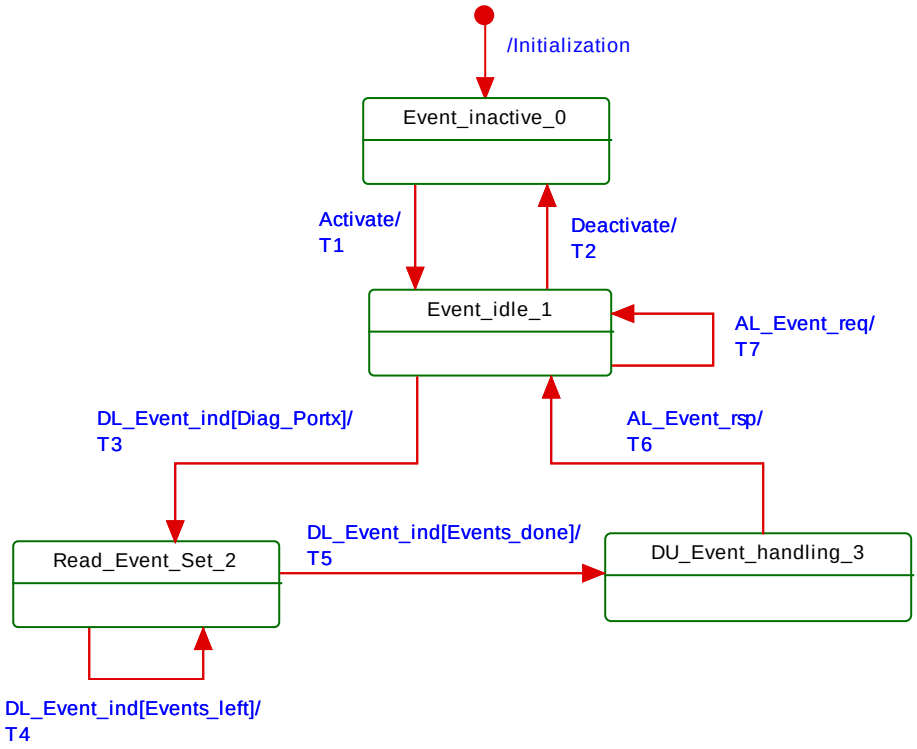


Figure 62 – Event state machine of the Master AL

Table 74 specifies the states and transitions of the Event state machine of the Master application layer.

Table 74 – State and transitions of the Event state machine of the Master AL

STATE NAME	STATE DESCRIPTION
Event_inactive_0	The AL Event handling of the Master is inactive.
Event_idle_1	The Master AL is ready to accept DL_Events (diagnosis information) from the DL.
Read_Event_Set_2	The Master AL received a DL_Event_ind with diagnosis information. After this first DL_Event.ind, the AL collects the complete set (1 to 6) of DL_Events of the current EventTrigger (see 11.5).
DU_Event_handling_3	The Master AL remains in this state as long as the Diagnosis Unit (see 11.5) did not acknowledge the AL_Event.ind.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	0	-
T3	1	2	-
T4	2	2	-
T5	2	3	AL_Event.ind
T6	3	1	DL_EventConf.req
T7	1	1	AL_Event.ind

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
Diag_Portx	Bool	Event set contains diagnosis information with details.
Events_done	Bool	Event set is processed.
Events_left	Bool	Event set not yet completed.

8.3.3.2 Event state machine of the Device AL

Figure 63 shows the Event state machine of the Device application layer.

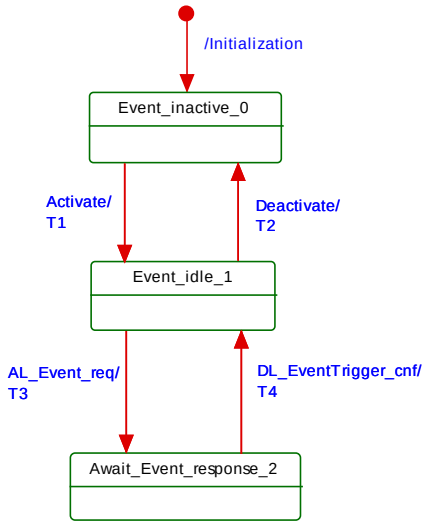


Figure 63 – Event state machine of the Device AL

Table 75 specifies the states and transitions of the Event state machine of the Device application layer.

Table 75 – State and transitions of the Event state machine of the Device AL

STATE NAME	STATE DESCRIPTION
Event_inactive_0	The AL Event handling of the Device is inactive.
Event_idle_1	The Device AL is ready to accept AL_Events (diagnosis information) from the technology specific Device applications for the transfer to the DL. The Device applications can create new Events during this time.
Await_event_response_2	The Device AL propagated an AL_Event with diagnosis information and waits on a DL_EventTrigger confirmation of the DL. The Device AL shall not accept any new AL_Event during this time.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	0	-
T3	1	2	An AL_Event request triggers a DL_Event and the corresponding DL_EventTrigger service. The DL_Event carries the diagnosis information from AL to DL. The DL_EventTrigger sets the Event flag within the cyclic data exchange (see A.1.5)
T4	2	1	A DL_EventTrigger confirmation triggers an AL_Event confirmation.

INTERNAL ITEMS	TYPE	DEFINITION
none		

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.3.3.3 Single Event scheduling

Figure 64 shows how a single Event from a Device is processed, in accordance with the relevant state machines.

- The Device application creates an Event request (Step 1), which is passed from the AL to the DL and buffered within the Event memory (see Table 56).
- The Device AL activates the EventTrigger service to raise the Event flag, which causes the Master to read the Event from the Event memory.
- The Master then propagates this Event to the gateway application (Step 2), and waits for an Event acknowledgement.
- Once the Event acknowledgement is received (Step 3), it is indicated to the Device by writing to the StatusCode (Step 4).
- The Device confirms the original Event request to its application (Step 5), which may now initiate a new Event request.

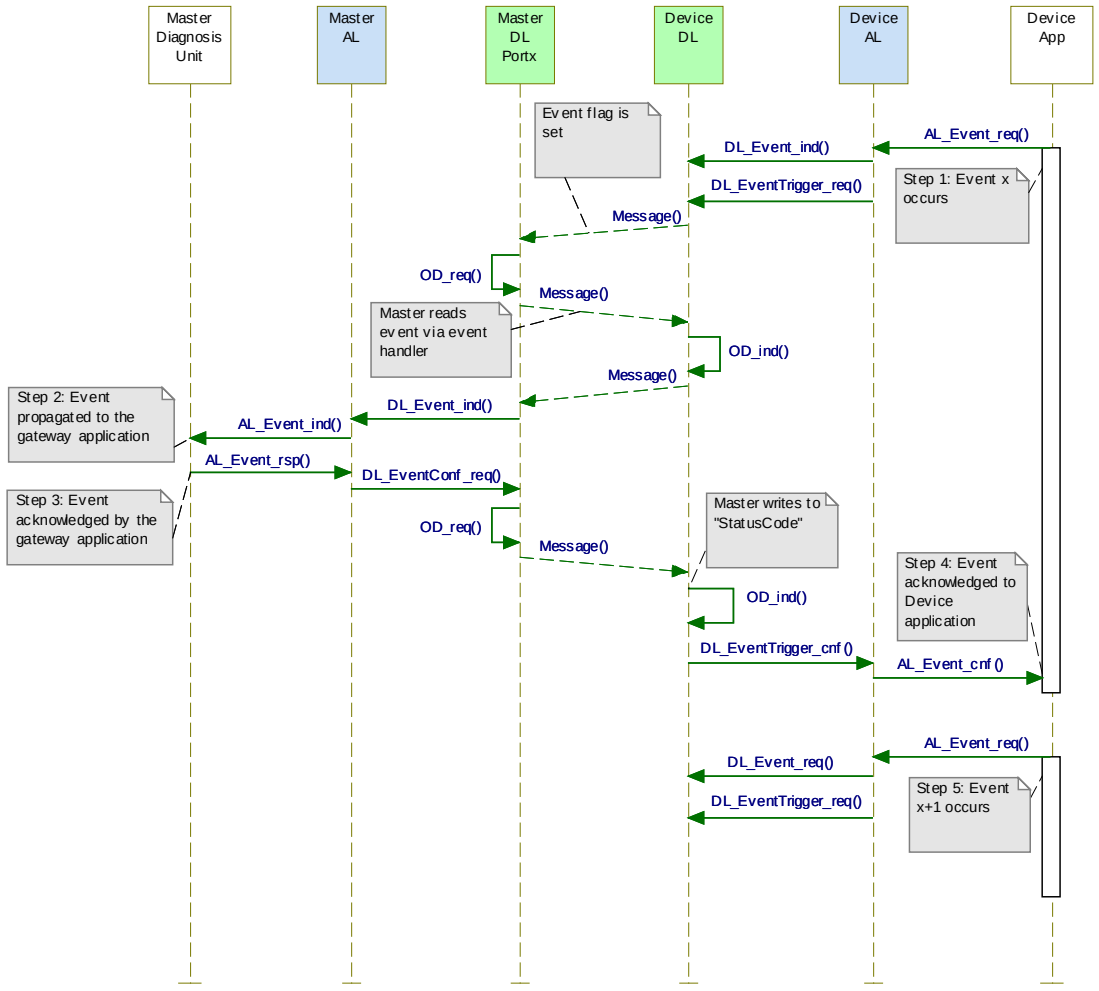


Figure 64 – Single Event scheduling

8.3.3.4 Multi Event transport (legacy Devices only)

Besides the method specified in 8.3.3.3 in which each single Event is conveyed through the layers and acknowledged by the gateway application, all Masters shall support a so-called "multi Event transport" which allows up to 6 Events to be transferred at a time. The Master AL transfers the Event set as a single diagnosis indication to the gateway application and returns a single acknowledgement for the entire set to the legacy Device application.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 64 also applies for the multi Event transport, except that this transport uses one DL_Event indication for each Event memory slot, and a single AL_Event indication for the entire Event set.

One AL_Event.req carries up to 6 Events and one AL_Event.ind indicates up to 6 pending Events. AL_Event.rsp and AL_Event.cnf refer to the indicated entire Event set.

8.3.4 Process Data cycles

Figure 65 and Figure 66 demonstrate complete interactions between Master and Device for output and input Process Data use cases.

Figure 65 demonstrates how the AL and DL services of Master and Device are involved in the cyclic exchange of output Process Data. The Device application is able to acquire the current values of output PD via the AL_GetOutput service.

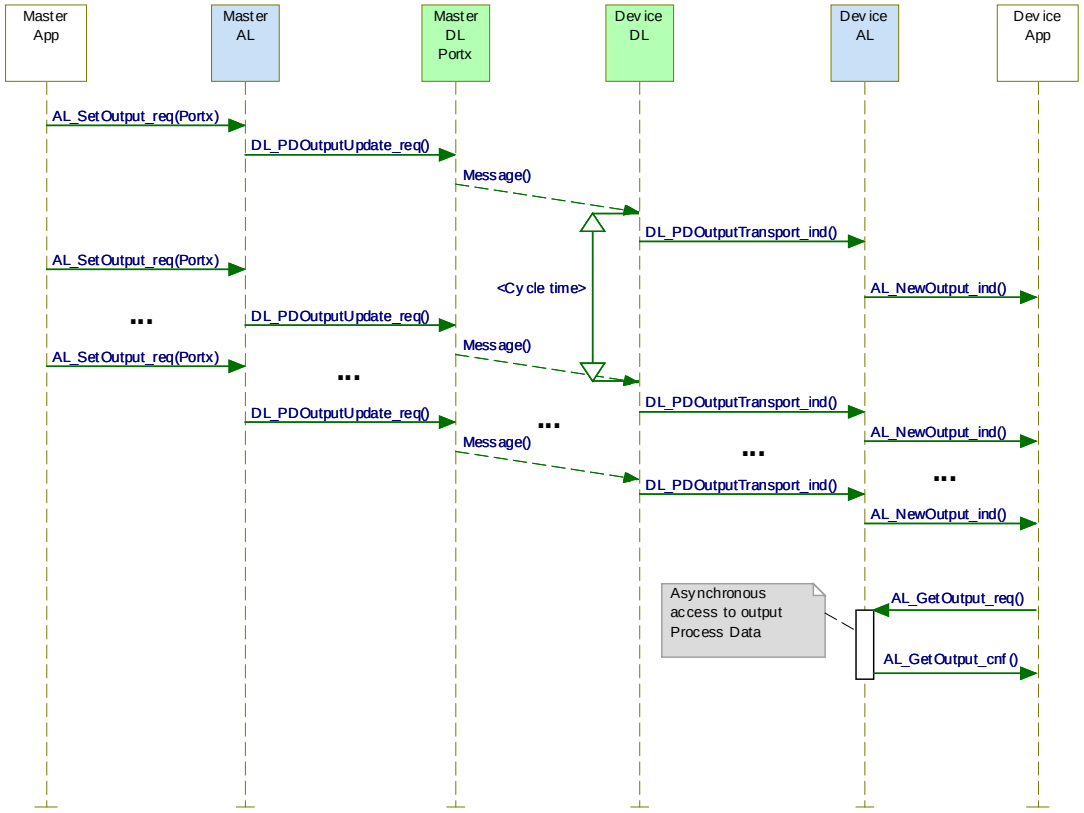


Figure 65 – Sequence diagram for output Process Data

Figure 66 demonstrates how the AL and DL services of Master and Device are involved in the cyclic exchange of input Process Data. The Master application is able to acquire the current values of input PD via the AL_GetInput service.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

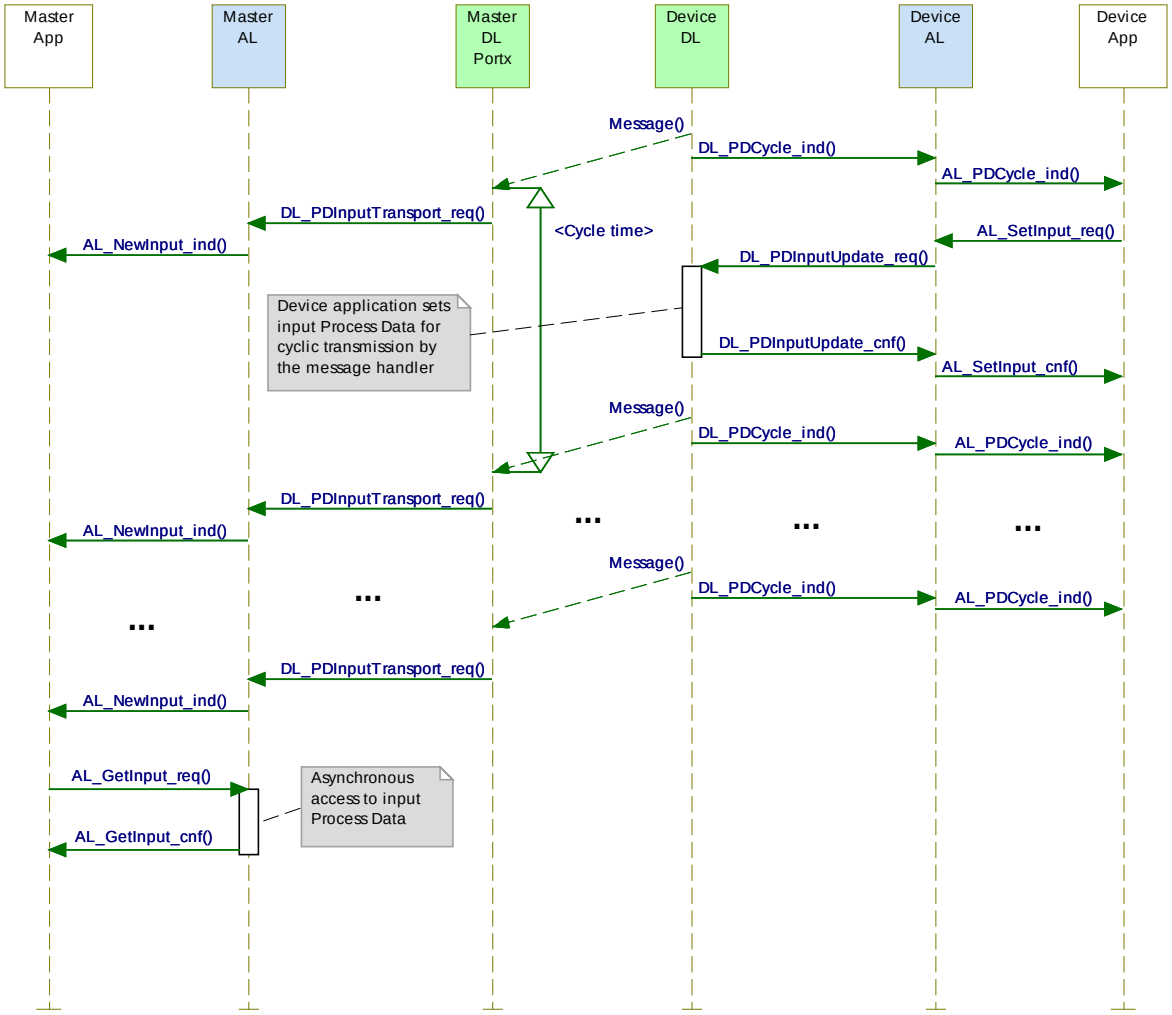


Figure 66 – Sequence diagram for input Process Data

9 System management (SM)

9.1 General

The SDCI system management is responsible for the coordinated startup of the ports within the Master and the corresponding operations within the connected Devices. The difference between the SM of the Master and the Device is more significant than with the other layers. Consequently, the structure of Clause 9 separates the services and protocols of Master and Device.

9.2 System management of the Master

9.2.1 Overview

The Master system management services are used to set up the Master ports and the system for all possible operational modes.

The Master SM adjusts ports through:

- establishing the required communication protocol revision;
- checking the Device compatibility (actual Device identifications match expected values);
- adjusting adequate Master M-sequence types and MasterCycleTimes.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

For this it uses the following services shown in Figure 67.

- SM_SetPortConfig transfers the necessary Device parameters (configuration data) from Configuration Management (CM) to System Mangement (SM). The port is then started implicitly.
- SM_PortMode reports the positive result of the port setup back to CM in case of correct port setup and inspection. It reports the negative result back to CM via corresponding "errors" in case of mismatching revisions and incompatible Devices.
- SM_GetPortConfig reads the actual and effective parameters.
- SM_Operate switches the ports into the "OPERATE" mode.

Figure 67 provides an overview of the structure and services of the Master system management.

The Master system management needs one application layer service (AL_Read) to acquire data (identification parameter) from special Indices for inspection.

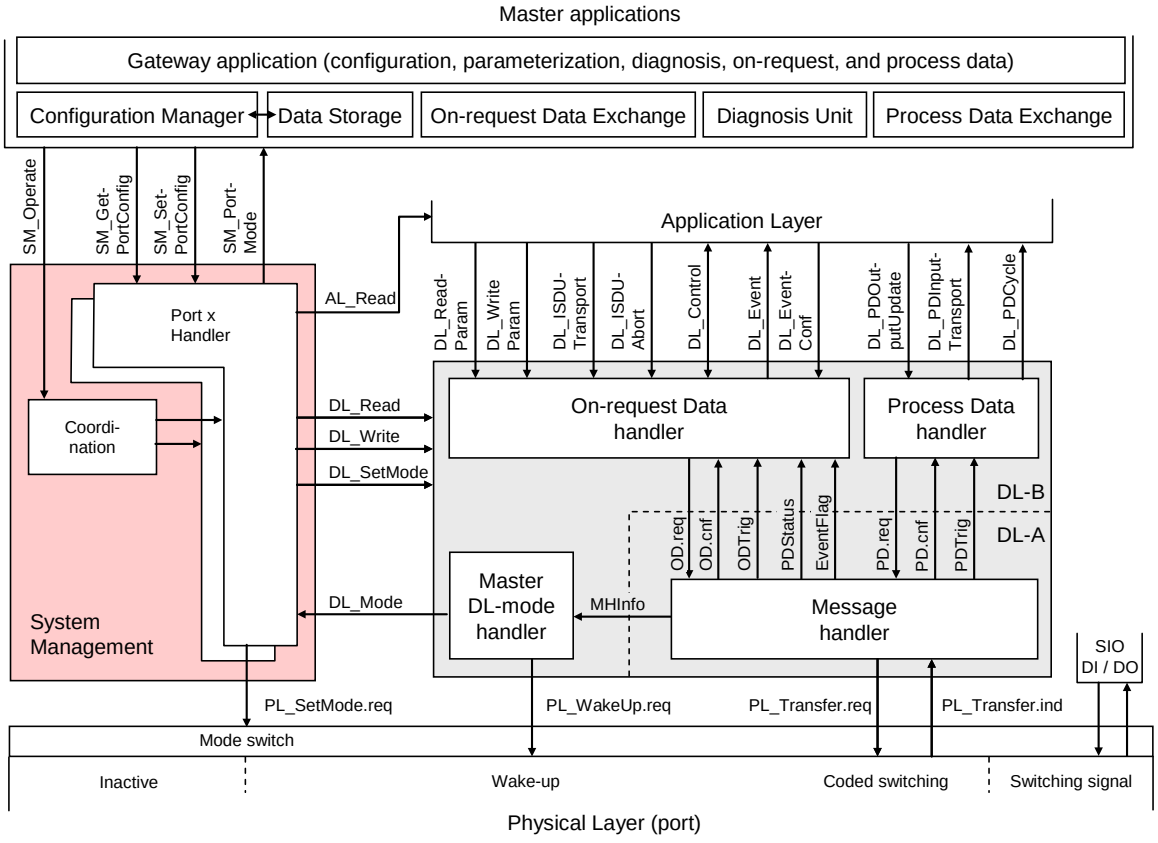


Figure 67 – Structure and services of the Master system management

Figure 68 demonstrates the actions between the layers Master application (Master App), Configuration Management (CM), System Management (SM), Data Link (DL) and Application Layer (AL) for the startup use case of a particular port.

This particular use case is characterized by the following statements:

- the Device for the available configuration is connected and inspection is successful;
- the Device uses the correct protocol version according to this specification;
- the configured InspectionLevel is "type compatible" (SerialNumber is read out of the Device and not checked).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Dotted arrows in Figure 68 represent response services to an initial service.

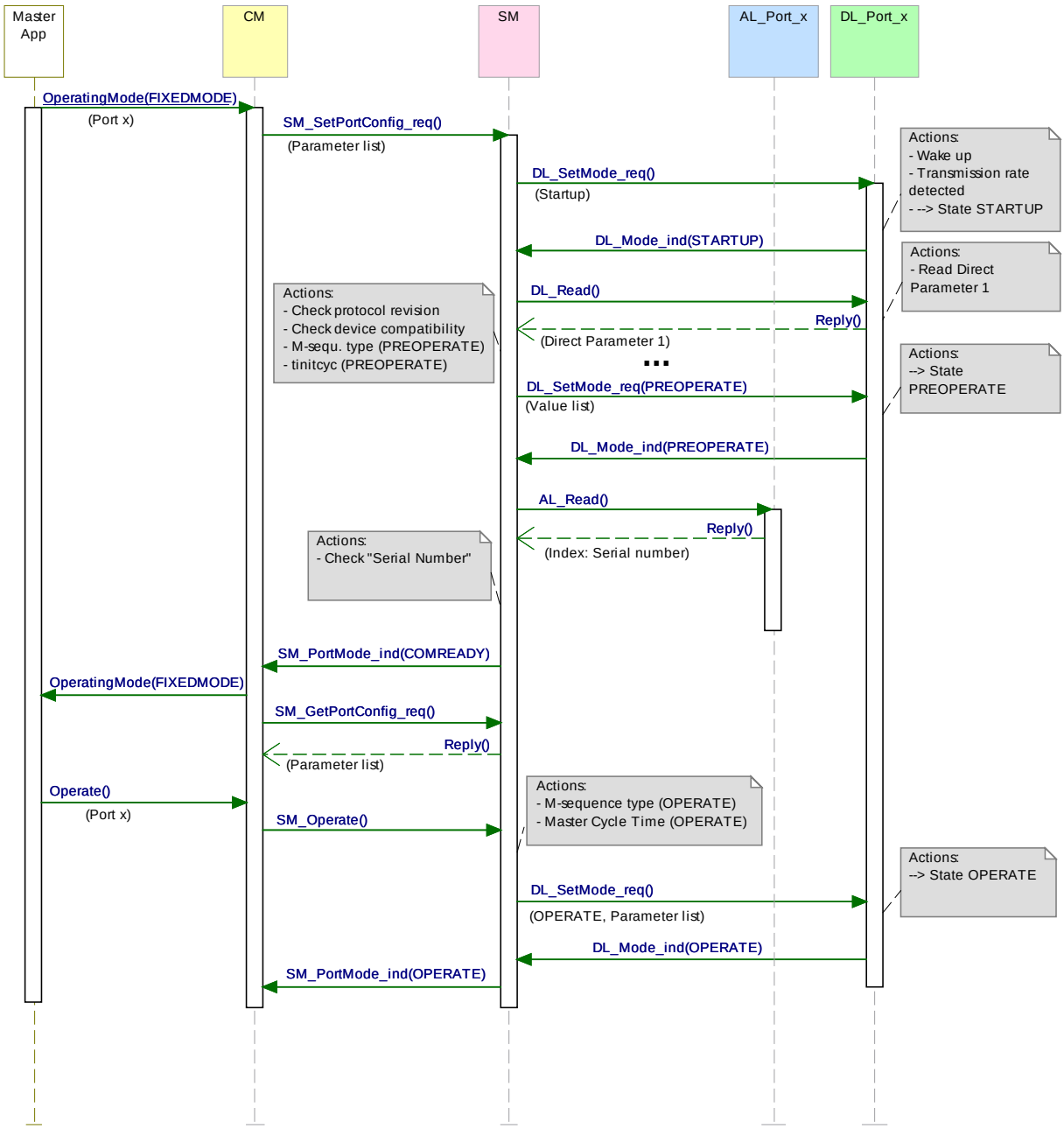


Figure 68 – Sequence chart of the use case "port x setup"

9.2.2 SM Master services

9.2.2.1 Overview

System management provides the SM Master services to the user via its upper interface. Table 76 lists the assignment of the Master to its role as initiator or receiver for the individual SM services.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 76 – SM services within the Master

Service name	Master
SM_SetPortConfig	R
SM_GetPortConfig	R
SM_PortMode	I
SM_Operate	R
Key (see 3.3.4) I Initiator of service R Receiver (Responder) of service	

9.2.2.2 SM_SetPortConfig

The SM_SetPortConfig service is used to set up the requested Device configuration. The parameters of the service primitives are listed in Table 77.

Table 77 – SM_SetPortConfig

Parameter name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Port Number		M
Result (-)		S
Port Number		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured port and Device parameters of a Master port.

Parameter type: Record

Record Elements:

Port Number
This parameter contains the port number

ConfiguredCycleTime
This parameter contains the requested cycle time for the OPERATE mode

Permitted values:
0 (FreeRunning)
Time (see Table B.3)

TargetMode
This parameter indicates the requested operational mode of the port

Permitted values: INACTIVE, DI, DO, CFGCOM, AUTOCOM (see Table 79)

ConfiguredBaudrate:
This parameter indicates the requested transmission rate

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permitted values:
AUTO (Master accepts transmission rate found during "ESTABLISHCOM")
COM1 (transmission rate of COM1)
COM2 (transmission rate of COM2)
COM3 (transmission rate of COM3)

ConfiguredRevisionID (CRID):
Data length: 1 octet for the protocol version (see B.1.5)

InspectionLevel:
Permitted values: NO_CHECK, TYPE_COMP, IDENTICAL (see Table 78)

ConfiguredVendorID (CVID)
Data length: 2 octets

NOTE VendorIDs are assigned by the IO-Link consortium

ConfiguredDeviceID (CDID)
Data length: 3 octets

ConfiguredFunctionID (CFID)
Data length: 2 octets

ConfiguredSerialNumber (CSN)
Data length: up to 16 octets

Result (+):
This selection parameter indicates that the service has been executed successfully

Port Number
This parameter contains the port number

Result (-):
This selection parameter indicates that the service failed

Port Number
This parameter contains the port number

ErrorInfo
This parameter contains error information

Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)

Table 78 specifies the coding of the different inspection levels (values of the InspectionLevel parameter) (see 9.2.3.2 and 11.8.5).

Table 78 – Definition of the InspectionLevel (IL)

Parameter	InspectionLevel (IL)		
	NO_CHECK	TYPE_COMP	IDENTICAL
DeviceID (DID) (compatible)	-	Yes (RDID=CDID)	Yes (RDID=CDID)
VendorID (VID)	-	Yes (RVID=CVID)	Yes (RVID=CVID)
SerialNumber (SN)	-	-	Yes (RSN = CSN)

Table 79 specifies the coding of the different Target Modes.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 79 – Definitions of the Target Modes

Target Mode	Definition
CFGCOM	Device communicating in mode CFGCOM after successful inspection
AUTOCOM	Device communicating in mode AUTOCOM without inspection
INACTIVE	Communication disabled, no DI, no DO
DI	Port in digital input mode (SIO)
DO	Port in digital output mode (SIO)

CFGCOM is a Target Mode based on a user configuration (for example with the help of an IODD) and consistency checking of RID, VID, DID.

AUTOCOM is a Target Mode without configuration. That means no checking of CVID and CDID. The CRID is set to the highest revision the Master is supporting. AUTOCOM should only be selectable together with Inspection Level "NO_CHECK" (see Table 78).

9.2.2.3 SM_GetPortConfig

The SM_GetPortConfig service is used to acquire the real (actual) Device configuration. The parameters of the service primitives are listed in Table 80.

Table 80 – SM_GetPortConfig

Parameter name	.req	.cnf
Argument	M	
Port Number	M	
Result (+)		S(=)
Parameterlist		M
Result (-)		S(=)
Port Number		M
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Port Number
This parameter contains the port number

Result (+):
This selection parameter indicates that the service request has been executed successfully.

ParameterList
This parameter contains the configured port and Device parameter of a Master port.

Parameter type: Record

Record Elements:

PortNumber
This parameter contains the port number.

TargetMode

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

This parameter indicates the operational mode

Permitted values: INACTIVE, DI, DO, CFGCOM, AUTOCOM (see Table 79)

RealBaudrate
This parameter indicates the actual transmission rate

Permitted values:
COM1 (transmission rate of COM1)
COM2 (transmission rate of COM2)
COM3 (transmission rate of COM3)

RealCycleTime
This parameter contains the real (actual) cycle time

RealRevision (RRID)
Data length: 1 octet for the protocol version (see B.1.5)

RealVendorID (RVID)
Data length: 2 octets

NOTE VendorIDs are assigned by the IO-Link consortium

RealDeviceID (RDID)
Data length: 3 octets

RealFunctionID (RFID)
Data length: 2 octets

RealSerialNumber (RSN)
Data length: up to 16 octets

Result (-):
This selection parameter indicates that the service failed

Port Number
This parameter contains the port number

ErrorInfo
This parameter contains error information

Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)

All parameters shall be set to "0" if there is no information available.

9.2.2.4 SM_PortMode

The SM_PortMode service is used to indicate changes or faults of the local communication mode. These shall be reported to the Master application. The parameters of the service primitives are listed in Table 81.

Table 81 – SM_PortMode

Parameter name	.ind
Argument	M
Port Number	M
Mode	M

Argument
The service-specific parameters are transmitted in the argument.

Port Number
This parameter contains the port number

Mode

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permitted values:

INACTIVE	(Communication disabled, no DI, no DO)
DI	(Port in digital input mode (SIO))
DO	(Port in digital output mode (SIO))
COMREADY	(Communication established and inspection successful)
SM_OPERATE	(Port is ready to exchange Process Data)
COMLOST	(Communication failed, new wake-up procedure required)
REVISION_FAULT	(Incompatible protocol revision)
COMP_FAULT	(Incompatible Device or Legacy-Device according to the InspectionLevel)
SERNUM_FAULT	(Mismatching SerialNumber according to the InspectionLevel)

9.2.2.5 SM_Operate

The SM_Operate service prompts system management to calculate the MasterCycleTimes of the ports when they are acknowledged positively with Result (+). This service is effective on all the ports. The parameters of the service primitives are listed in Table 82.

Table 82 – SM_Operate

Parameter name	.req	.cnf
Result (+)		S
Result (-)		S
ErrorInfo		M

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:

TIMING_CONFLICT	(the requested combination of cycle times for the activated ports is not possible)

9.2.3 SM Master protocol

9.2.3.1 Overview

Due to the comprehensive configuration, parameterization, and operational features of SDCI the description of the behavior with the help of state diagrams becomes rather complex. Similar to the DL state machines 9.2.3 uses the possibility of submachines within the main state machines.

Comprehensive compatibility check methods are performed within the submachine states. These methods are indicated by "do *method*" fields within the state graphs, for example in Figure 70.

The corresponding decision logic is demonstrated via activity diagrams (see Figure 71, Figure 72, Figure 73, and Figure 76).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.2.3.2 SM Master state machine

Figure 69 shows the main state machine of the System Mangement Master. Two submachines for the compatibility and serial number check are specified in 9.2.3.3 and 9.2.3.4. In case of communication disruption the system management is informed via the service DL_Mode (COMLOST). Only the SM_SetPortConfig service allows reconfiguration of a port. The service SM_Operate (effective on all ports) causes no effect in any state except in state "wait_4".

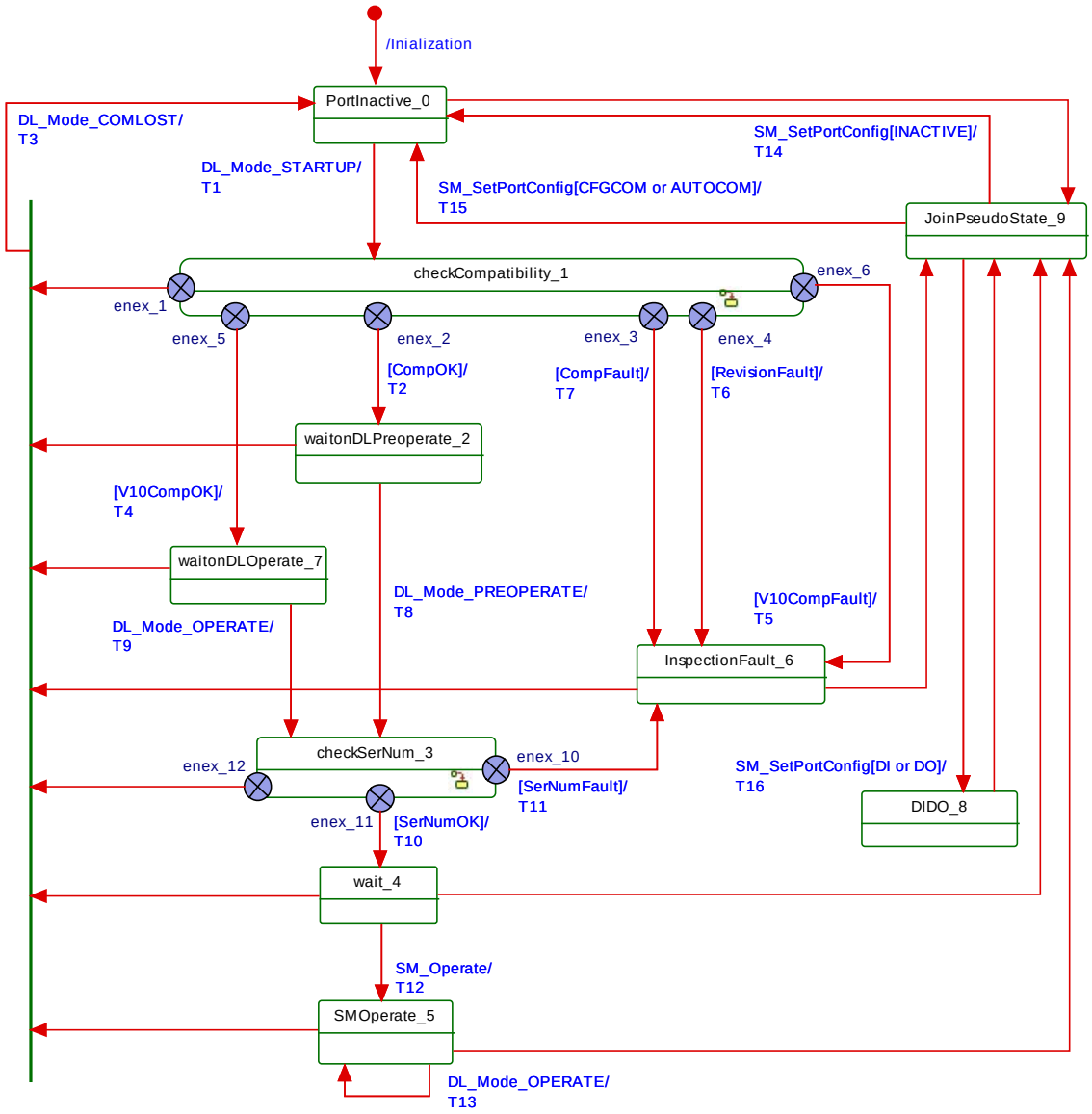


Figure 69 – Main state machine of the Master system management

Table 83 shows the state transition tables of the Master system management.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 83 – State transition tables of the Master system management

STATE NAME	STATE DESCRIPTION
PortInactive_0	No communication
CheckCompatibility_1	Port is started and revision and Device compatibility is checked. See Figure 70.
waitonDLPreoperate_2	Wait until the PREOPERATE state is established and all the On-Request handlers are started. Port is ready to communicate.
CheckSerNum_3	SerialNumber is checked depending on the InspectionLevel (IL). See Figure 75.
wait_4	Port is ready to communicate and waits on service SM_Operate from CM.
SM Operate_5	Port is in state OPERATE and performs cyclic Process Data exchange.
InspectionFault_6	Port is ready to communicate. However, cyclic Process Data exchange cannot be performed due to incompatibilities.
waitonDLOperate_7	Wait on the requested state OPERATE in case the Master is connected to a legacy Device. The SerialNumber can be read thereafter.
DIDO_8	Port will be switched into the DI or DO mode (SIO, no communication)
JoinPseudoState_9	This pseudo state is used instead of a UML join bar. It allows execution of individual SM_SetPortConfig services depending on the system status (INACTIVE, CFGCOM, AUTOCOM, DI, or DO)

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	CompRetry = 0
T2	1	2	DL_SetMode.req (PREOPERATE, ValueList)
T3	1,2,3,4,5, 6,7	0	DL_SetMode.req (INACTIVE) and SM_Mode.ind (COMLOST) due to communiacation fault
T4	1	7	DL_SetMode.req (OPERATE, ValueList)
T5	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (OPERATE, ValueList)
T6	1	6	SM_PortMode.ind (REVISION_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T7	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T8	2	3	-
T9	7	3	-
T10	3	4	SM_PortMode.ind (COMREADY)
T11	3	6	SM_PortMode.ind (SERNUM_FAULT)
T12	4	5	DL_SetMode.req (OPERATE, ValueList)
T13	5	5	-
T14	0,4,5,6,8	0	SM_PortMode.ind (INACTIVE), DL_SetMode.req (INACTIVE)
T15	0,4,5,6,8	0	DL_SetMode.req (STARTUP, ValueList), PL_SetMode.req (SDCI)
T16	0,4,5,6,8	8	PL_SetMode.req (SIO), SM_Mode.ind (DI or DO), DL_SetMode.req (INACTIVE)

INTERNAL ITEMS	TYPE	DEFINITION
CompOK	Bool	See Figure 73
CompFault	Bool	See Figure 73; error variable COMP_FAULT
RevisionFault	Bool	See Figure 71; error variable REVISION_FAULT
SerNumFault	Bool	See Figure 76; error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 76
V10CompFault	Bool	See Figure 72; error variable COMP_FAULT
V10CompOK	Bool	See Figure 72

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
INACTIVE	Variable	A target mode in service SM_SetPortConfig
CFGCOM, AUTOCOM	Variables	Target Modes in service SM_SetPortConfig

9.2.3.3 SM Master submachine "Check Compatibility"

Figure 70 shows the SM Master submachine checkCompatibility_1.

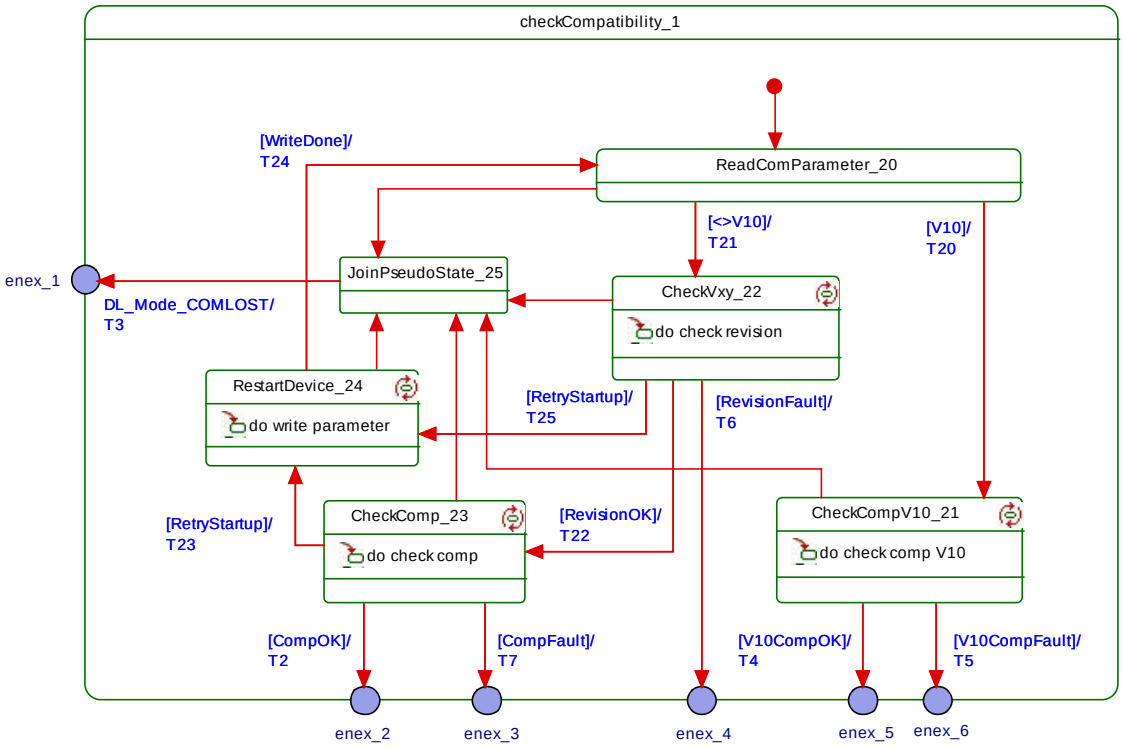


Figure 70 – SM Master submachine CheckCompatibility_1

Table 84 shows the state transition tables of the Master submachine checkCompatibility_1.

Table 84 – State transition tables of the Master submachine CheckCompatibility_1

STATE NAME	STATE DESCRIPTION
ReadComParameter_20	Acquires communication parameters from Direct Parameter Page 1 (0x02 to 0x06) via service DL_Read (see Table B.1).
CheckCompV10_21	Acquires identification parameters from Direct Parameter Page 1 (0x07 to 0x0D) via service DL_Read (see Table B.1). The configured InspectionLevel (IL) defines the decision logic of the subsequent compatibility check "CheckCompV10" with parameters RVID, RDID, and RFID according to Figure 72.
CheckVxy_22	A check is performed whether the configured revision (CRID) matches the real (actual) revision (RRID) according to Figure 71.
CheckComp_23	Acquires identification parameters from Direct Parameter Page 1 (0x07 to 0x0D) via service DL_Read (see Table B.1). The configured InspectionLevel (IL) defines the decision logic of the subsequent compatibility check "CheckComp" according to Figure 73.
RestartDevice_24	Writes the compatibility parameters configured protocol revision (CRID) and configured DeviceID (CDID) into the Device depending on the Target Mode of communication CFGCOM or AUTOCOM (see Table 79) according to Figure 74.
JoinPseudoState_25	This pseudo state is used instead of a UML join bar. No guards involved.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T20	20	21	-
T21	20	22	DL_Write (0x00, MCmd_MASTERIDENT), see Table B.2
T22	22	23	-
T23	23	24	-
T24	24	20	-
T25	22	24	CompRetry = CompRetry +1

INTERNAL ITEMS	TYPE	DEFINITION
CompOK	Bool	See Figure 73
CompFault	Bool	See Figure 73; error variable COMP_FAULT
RevisionFault	Bool	See Figure 71; error variable REVISION_FAULT
RevisionOK	Bool	See Figure 71
SerNumFault	Bool	See Figure 76; error variable SERNUM_FAULT
SerNumOK	Bool	See Figure 76
V10	Bool	Real protocol revision of connected Device is a legacy version (V1.0, see B.1.5)
<>V10	Bool	Real protocol revision of connected Device is in accordance with this standard
V10CompFault	Bool	See Figure 72; error variable COMP_FAULT
V10CompOK	Bool	See Figure 72
RetryStartup	Bool	See Figure 71 and Figure 73
CompRetry	Variable	Internal counter
WriteDone	Bool	Finalization of the restart service sequence
MCmd_XXXXXXX	Call	See Table 43

Some states contain complex logic to deal with the compatibility and validity checks. Figure 71 to Figure 74 are demonstrating the context.

Figure 71 shows the decision logic for the protocol revision check in state "CheckVxy". In case of configured Devices the following rule applies: if the configured revision (CRID) and the real revision (RRID) do not match, the CRID will be transmitted to the Device. If the Device does not accept, the Master returns an indication via the SM_Mode service with REV_FAULT.

In case of not configured Devices the operational mode AUTOCOM shall be used. See 9.2.2.2 and 9.2.2.3 for the parameter name abbreviations.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

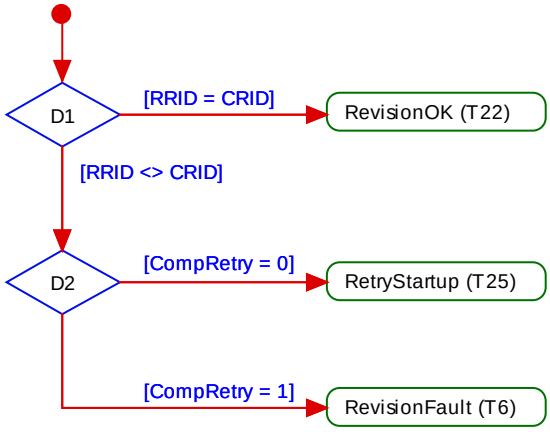
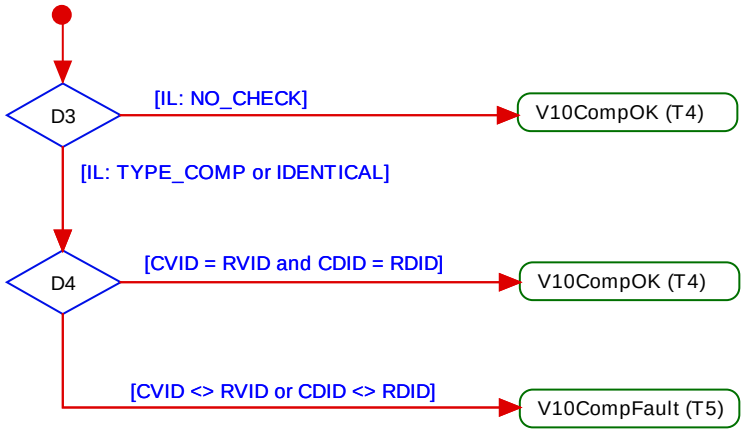


Figure 71 – Activity for state "CheckVxy"

Figure 72 shows the decision logic for the legacy compatibility check in state "CheckCompV10".

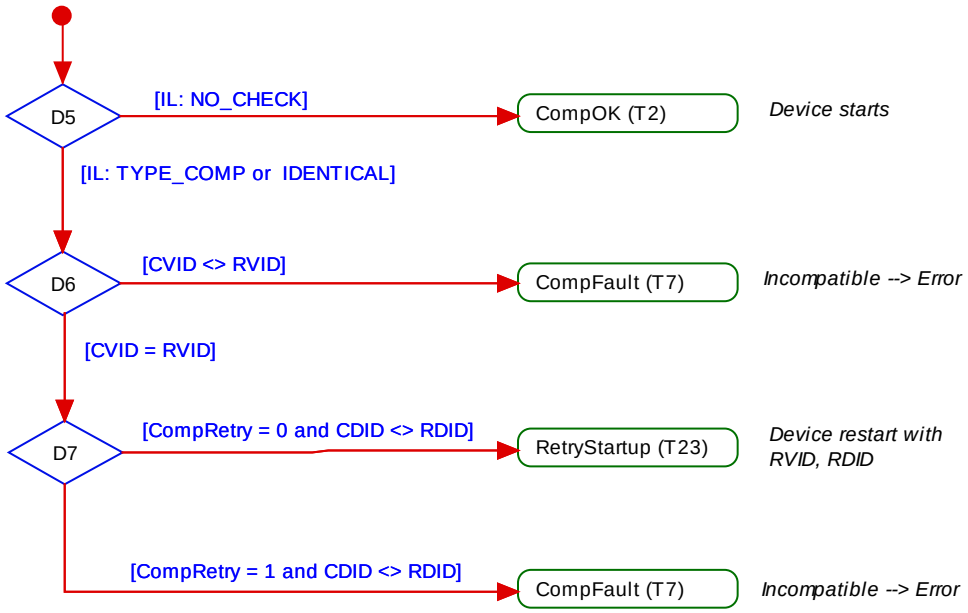


Key:
IL = Inspection level

Figure 72 – Activity for state "CheckCompV10"

Figure 73 shows the decision logic for the compatibility check in state "CheckComp".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Key:
IL = Inspection level

Figure 73 – Activity for state "CheckComp"

Figure 74 shows the activity (write parameter) in state "RestartDevice".

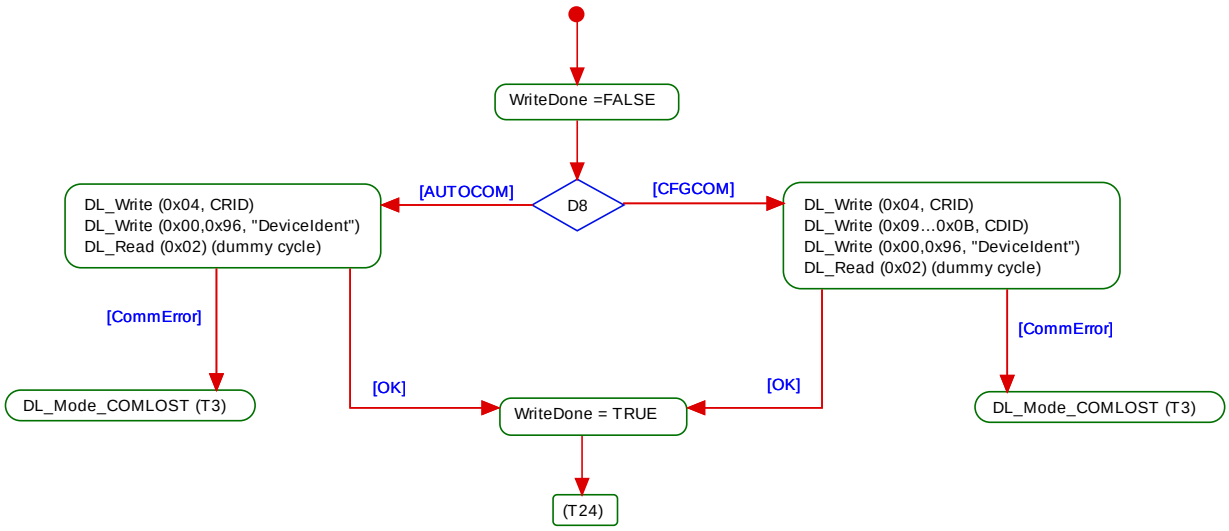


Figure 74 – Activity (write parameter) in state "RestartDevice"

9.2.3.4 SM Master submachine "Check serial number"

Figure 75 shows the SM Master submachine "checkSerNum_3". This check is mandatory.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

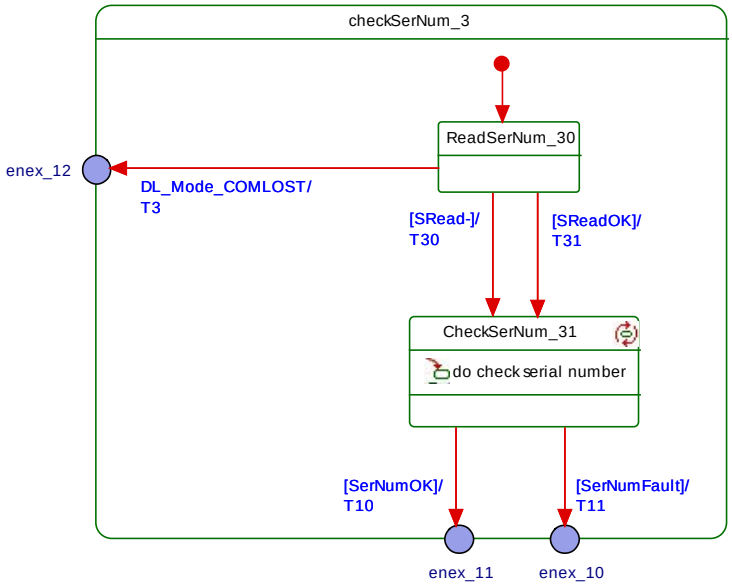


Figure 75 – SM Master submachine CheckSerNum_3

Table 85 shows the state transition tables of the Master submachine CheckSerNum_3

Table 85 – State transition tables of the Master submachine CheckSerNum_3

STATE NAME	STATE DESCRIPTION
ReadSerNum_30	Acquires the SerialNumber from the Device via AL_Read.req (Index: 0x0015). A positive response (AL_Read(+)) leads to SReadOK = true. A negative response (AL_Read(-)) leads to SRead- = true.
CheckSerNum_31	The configured (CSN) and the real (RSN) SerialNumber are checked depending on the InspectionLevel (IL) according to Figure 76.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T30	40	41	
T31	40	41	

INTERNAL ITEMS	TYPE	DEFINITION
SRead-	Bool	Negative response of service AL_Read (Index 0x0015)
SReadOK	Bool	SerialNumber read correctly
SERNumOK	Bool	See Figure 76
SERNumFault	Bool	See Figure 76

Figure 76 shows the decision logic (activity) for the state CheckSerNum_3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

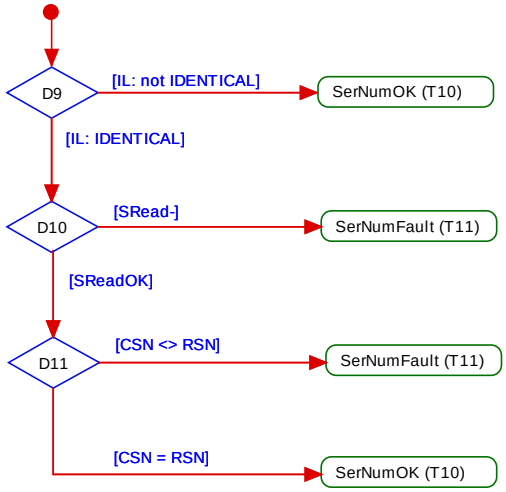


Figure 76 – Activity (check SerialNumber) for state CheckSerNum_3

9.2.3.5 Rules for the usage of M-sequence types

The System management is responsible for setting up the correct M-sequence types. This occurs after the check compatibility actions (transition to PREOPERATE) and before the transition to OPERATE.

Different M-sequence types shall be used within the different operational states (see A.2.6). For example, when switching to the OPERATE state the M-sequence type relevant for cyclic operation shall be used. The M-sequence type to be used in operational state OPERATE is determined by the size of the input and output Process Data. The available M-sequence types in the three modes STARTUP, PREOPERATE, and OPERATE and the corresponding coding of the parameter M-sequence Capability are specified in A.2.6. The input and output data formats shall be acquired from the connected Device in order to adjust the M-sequence type. It is mandatory for a Master to implement all the specified M-sequence types in A.2.6.

9.3 System management of the Device

9.3.1 Overview

Figure 77 provides an overview of the structure and services of the Device system management.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

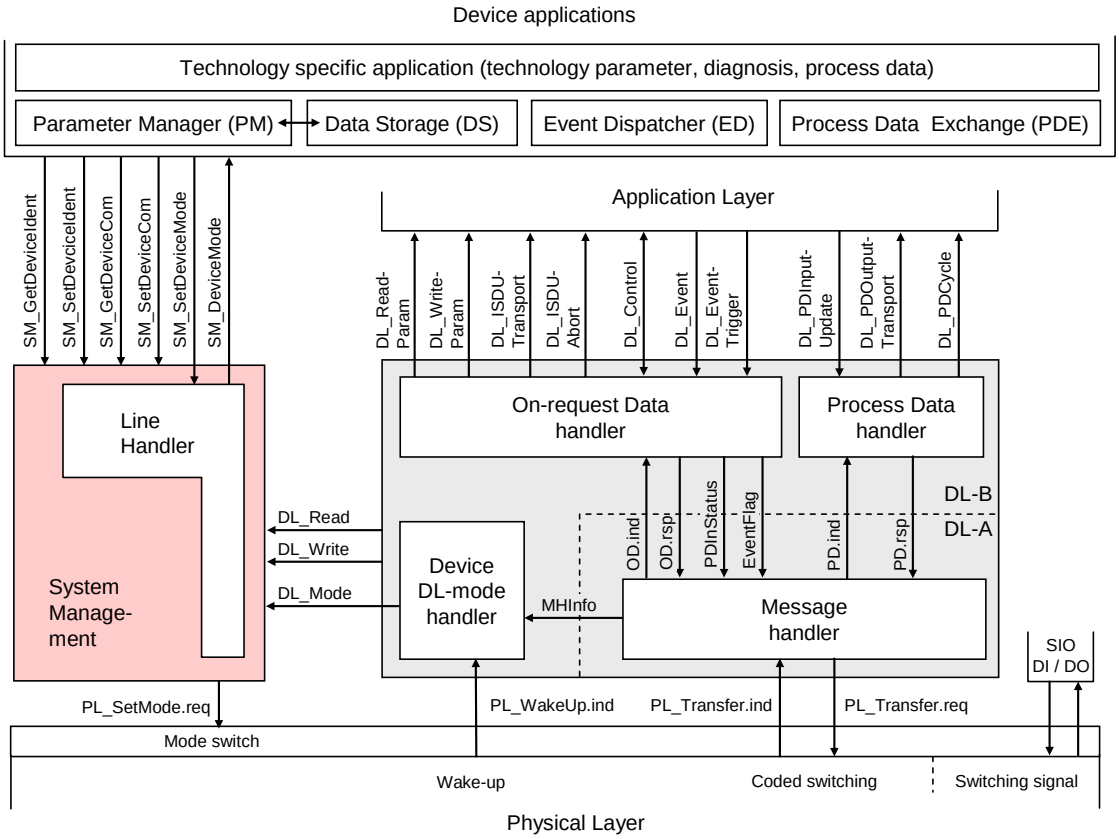


Figure 77 – Structure and services of the system management (Device)

The System Management (SM) of the Device provides the central controlling instance via the Line Handler through all the phases of initialization, default state (SIO), communication startup, communication, and fall-back to SIO mode.

The Device SM interacts with the PL to establish the necessary line driver and receiver adjustments (see Figure 15), with the DL to get the necessary information from the Master (wake-up, transmission rates, a.o.) and with the Device applications to ensure the Device identity and compatibility (identification parameters).

The transitions between the line handler states (see Figure 79) are initiated by the Master port activities (wake-up and communication) and triggered through the Device Data Link Layer via the DL_Mode indications and DL_Write requests (commands).

The SM provides the Device identification parameters through the Device applications interface.

The sequence chart in Figure 78 demonstrates a typical Device sequence from initialization to default SIO mode and via wake-up request from the Master to final communication. The sequence chart is complemented by the use case of a communication error such as T_{DSIO} expired, or communication fault, or a request from Master such as Fallback (caused by Event).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

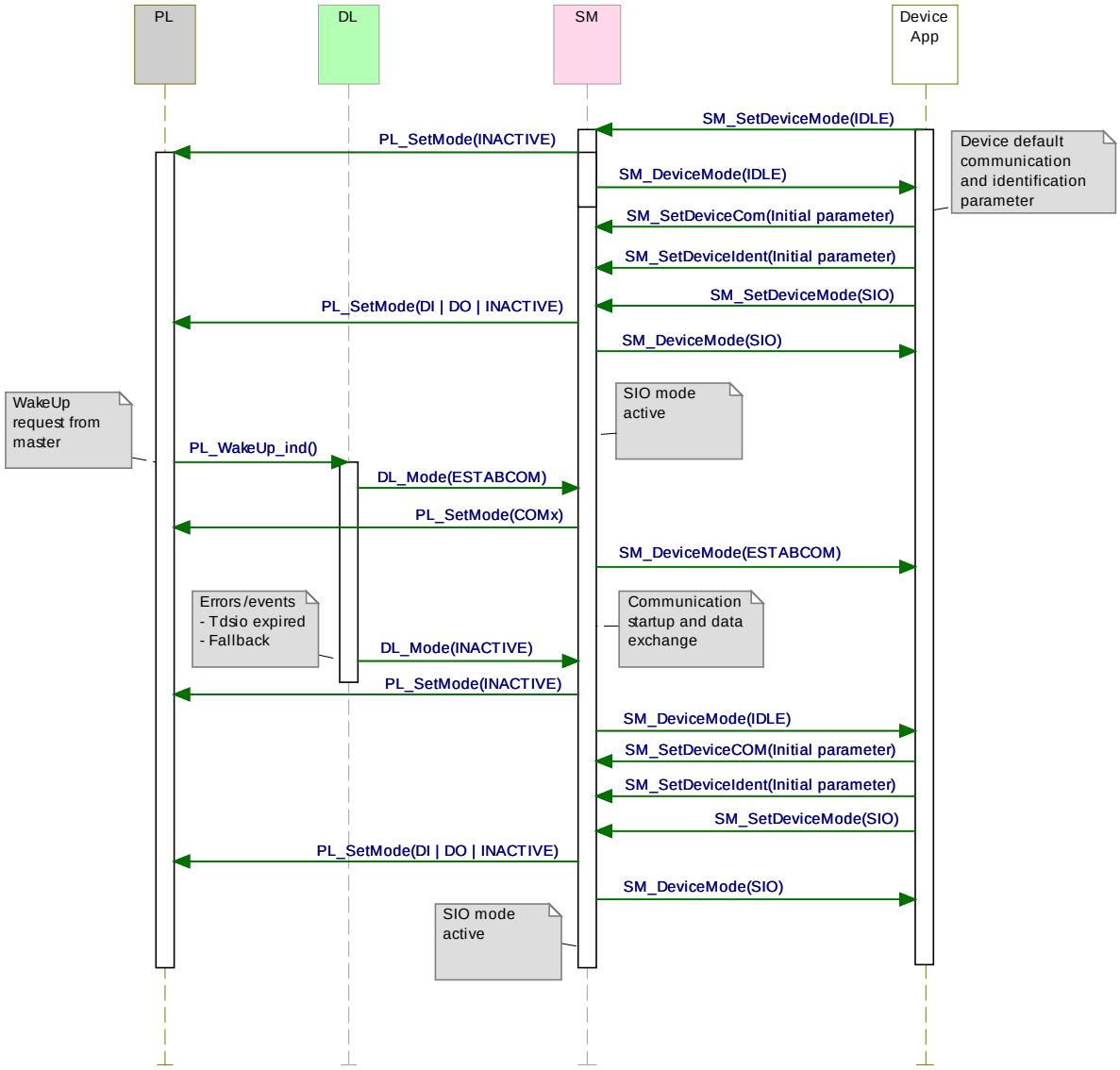


Figure 78 – Sequence chart of the use case "INACTIVE – SIO – SDCI – SIO"

The SM services shown in Figure 78 are specified in 9.3.2.

9.3.2 SM Device services

9.3.2.1 Overview

Subclause 9.3.2 describes the services the Device system management provides to its applications as shown in Figure 77.

Table 86 lists the assignment of the Device to its role as initiator or receiver for the individual system management service.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 86 – SM services within the Device

Service name	Device
SM_SetDeviceCom	R
SM_GetDeviceCom	R
SM_SetDeviceIdent	R
SM_GetDeviceIdent	R
SM_SetDeviceMode	R
SM_DeviceMode	I
Key (see 3.3.4) I Initiator of service R Receiver (Responder) of service	

9.3.2.2 SM_SetDeviceCom

The SM_SetDeviceCom service is used to configure the communication properties supported by the Device in the system management. The parameters of the service primitives are listed in Table 87.

Table 87 – SM_SetDeviceCom

Parameter name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured communication parameters for a Device.

Parameter type: Record

Record Elements:

SupportedSIOMode
This parameter indicates the SIO mode supported by the Device.

Permitted values:
INACTIVE (C/Q line in high impedance),
DI (C/Q line in digital input mode),
DO (C/Q line in digital output mode),

SupportedTransmissionrate
This parameter indicates the transmission rates supported by the Device.

Permitted values:
COM1 (transmission rate of COM1)
COM2 (transmission rate of COM2)
COM3 (transmission rate of COM3)

MinCycleTime
This parameter contains the minimum cycle time supported by the Device (see B.1.3).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

M-sequence Capability
This parameter indicates the capabilities supported by the Device (see B.1.4):
- ISDU support
- OPERATE M-sequence types
- PREOPERATE M-sequence types

RevisionID (RID)
This parameter contains the protocol revision (see B.1.5) supported by the Device.

ProcessDataIn
This parameter contains the length of PD to be sent to the Master (see B.1.6).

ProcessDataOut
This parameter contains the length of PD to be sent by the Master (see B.1.7).

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
PARAMETER_CONFLICT (consistency of parameter set violated)

9.3.2.3 SM_GetDeviceCom

The SM_GetDeviceCom service is used to read the current communication properties from the system management. The parameters of the service primitives are listed in Table 88.

Table 88 – SM_GetDeviceCom

Parameter name	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Result (+):
This selection parameter indicates that the service has been executed successfully.

ParameterList
This parameter contains the configured communication parameter for a Device.

Parameter type: Record

Record Elements:

CurrentMode
This parameter indicates the current SIO or Communication Mode by the Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permitted values:
INACTIVE (C/Q line in high impedance)
DI (C/Q line in digital input mode)
DO (C/Q line in digital output mode)
COM1 (transmission rate of COM1)
COM2 (transmission rate of COM2)
COM3 (transmission rate of COM3)

MasterCycleTime
This parameter contains the MasterCycleTime to be set by the Master system management (see B.1.3). This parameter is only valid in the state SM_Operate.

M-sequence Capability
This parameter indicates the current M-sequence capabilities configured in the system management of the Device (see B.1.4).
- ISDU support
- OPERATE M-sequence types
- PREOPERATE M-sequence types

RevisionID (RID)
This parameter contains the current protocol revision (see B.1.5) within the system management of the Device.

ProcessDataIn
This parameter contains the current length of PD to be sent to the Master (see B.1.6).

ProcessDataOut
This parameter contains the current length of PD to be sent by the Master (see B.1.7).

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (service unavailable within current state)

9.3.2.4 SM_SetDeviceIdent

The SM_SetDeviceIdent service is used to configure the Device identification data in the system management. The parameters of the service primitives are listed in Table 89.

Table 89 – SM_SetDeviceIdent

Parameter name	.req	.cnf
Argument	M	
ParameterList	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

ParameterList
This parameter contains the configured identification parameter for a Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Parameter type: Record

Record Elements:

VendorID (VID)
This parameter contains the VendorID assigned to a Device (see B.1.8)

Data length: 2 octets

DeviceID (DID)
This parameter contains one of the assigned DeviceIDs (see B.1.9)

Data length: 3 octets

FunctionID (FID)
This parameter contains one of the assigned FunctionIDs (see B.1.10).

Data length: 2 octets

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (service unavailable within current state)
PARAMETER_CONFLICT (consistency of parameter set violated)

9.3.2.5 SM_GetDeviceIdent

The SM_GetDeviceIdent service is used to read the Device identification parameter from the system management. The parameters of the service primitives are listed in Table 90.

Table 90 – SM_GetDeviceIdent

Parameter name	.req	.cnf
Argument	M	
Result (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Result (+):
This selection parameter indicates that the service has been executed successfully.

ParameterList
This parameter contains the configured communication parameters of the Device.

Parameter type: Record

Record Elements:

VendorID (VID)
This parameter contains the actual VendorID of the Device (see B.1.8)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Data length: 2 octets

DeviceID (DID)
This parameter contains the actual DeviceID of the Device (see B.1.9)

Data length: 3 octets

FunctionID (FID)
This parameter contains the actual FunctionID of the Device (see B.1.10).

Data length: 2 octets

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (service unavailable within current state)

9.3.2.6 SM_SetDeviceMode

The SM_SetDeviceMode service is used to set the Device into a defined operational state during initialization. The parameters of the service primitives are listed in Table 91.

Table 91 – SM_SetDeviceMode

Parameter name	.req	.cnf
Argument	M	
Mode	M	
Result (+)		S
Result (-)		S
ErrorInfo		M

Argument
The service-specific parameters are transmitted in the argument.

Mode
Permitted values:
IDLE (Device changes to waiting for configuration)
SIO (Device changes to the mode defined in service "SM_SetDeviceCom")

Result (+):
This selection parameter indicates that the service has been executed successfully.

Result (-):
This selection parameter indicates that the service failed.

ErrorInfo
This parameter contains error information.

Permitted values:
STATE_CONFLICT (service unavailable within current state)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.3.2.7 SM_DeviceMode

The SM_DeviceMode service is used to indicate changes of communication states to the Device application. The parameters of the service primitives are listed in Table 92.

Table 92 – SM_DeviceMode

Parameter name	.ind
Argument	M
Mode	M

Argument
The service-specific parameters are transmitted in the argument.

Mode
Permitted values:

IDLE	(Device changed to waiting for configuration)
SIO	(Device changed to the mode defined in service "SM_SetDeviceCom")
ESTABCOM	(Device changed to the SM mode "SM_ComEstablish")
COM1	(Device changed to the COM1 mode)
COM2	(Device changed to the COM2 mode)
COM3	(Device changed to the COM3 mode)
STARTUP	(Device changed to the STARTUP mode)
IDENT_STARTUP	(Device changed to the SM mode "SM_IdentStartup")
IDENT_CHANGE	(Device changed to the SM mode "SM_IdentCheck")
PREOPERATE	(Device changed to the PREOPERATE mode)
OPERATE	(Device changed to the OPERATE mode)

9.3.3 SM Device protocol

9.3.3.1 Overview

The behaviour of the Device is mainly driven by Master messages.

9.3.3.2 SM Device state machine

Figure 79 shows the SM line handler state machine of the Device. It is triggered by the DL_Mode handler and the Device application. It evaluates the different communication phases during startup and controls the line state of the Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

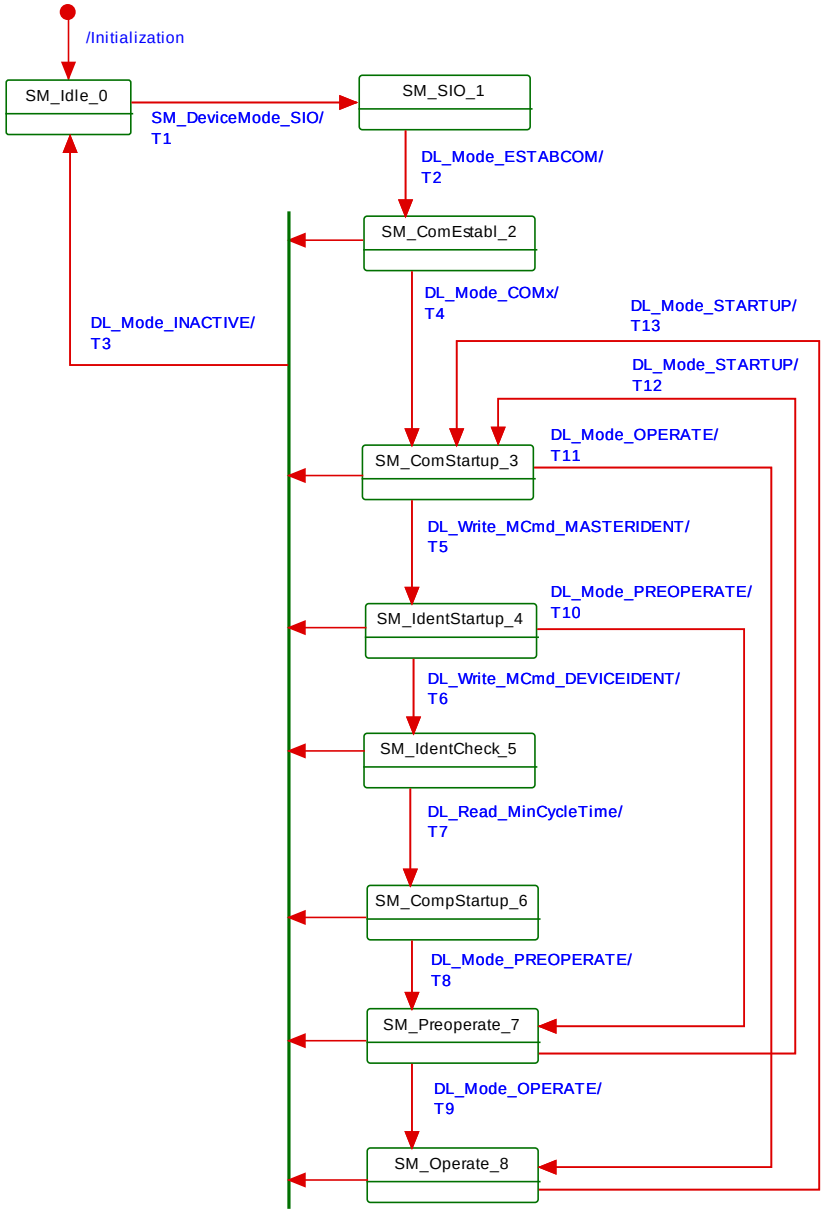


Figure 79 – State machine of the Device system management

Table 93 specifies the individual states and the actions within the transitions.

Table 93 – State transition tables of the Device system management

STATE NAME	STATE DESCRIPTION
SM_Idle_0	In SM_Idle the SM is waiting for configuration by the Device application and to be set to SIO mode. The state is left on receiving a SM_SetDeviceMode(SIO) request from the Device application The following sequence of services shall be executed between Device application and SM. Invoke SM_SetDeviceCom(initial parameter list) Invoke SM_SetDeviceIdent(VID, initial DID, FID)
SM_SIO_1	In SM_SIO the SM Line Handler is remaining in the default SIO mode. The Physical Layer is set to the SIO mode characteristics defined by the Device application via the SetDeviceMode service. The state is left on receiving a DL_Mode(ESTABCOM) indication.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

STATE NAME	STATE DESCRIPTION
SM_ComEstablish_2	In SM_ComEstablish the SM is waiting for the communication to be established in the Data Link Layer. The state is left on receiving a DL_Mode(INACTIVE) or a DL_Mode(COMx) indication, where COMx may be any of COM1, COM2 or COM3.
SM_ComStartup_3	In SM_ComStartup the communication parameter (Direct Parameter page 1, addresses 0x02 to 0x06) are read by the Master SM via DL_Read requests. The state is left upon reception of a DL_Mode(INACTIVE), a DL_Mode(OPERATE) indication (legacy Master only), or a DL_Write(MCmd_MASTERIDENT) request (Master in accordance with this standard).
SM_IdentStartup_4	In SM_IdentStartup the identification data (VID, DID, FID) are read and verified by the Master. In case of incompatibilities the Master SM writes the supported SDCI Revision (RID) and configured DeviceID (DID) to the Device. The state is left upon reception of a DL_Mode(INACTIVE), a DL_Mode(PREOPERATE) indication (compatibility check passed), or a DL_Write(MCmd_DEVICEIDENT) request (new compatibility requested).
SM_IdentCheck_5	In SM_IdentCheck the SM waits for new initialization of communication and identification parameters. The state is left on receiving a DL_Mode(INACTIVE) indication or a DL_Read(Direct Parameter page 1, addresses 0x02 = "MinCycleTime") request. Within this state the Device application shall check the RID and DID parameters from the SM and set these data to the supported values. Therefore the following sequence of services shall be executed between Device application and SM. Invoke SM_GetDeviceCom(configured RID, parameter list) Invoke SM_GetDeviceIdent(configured DID, parameter list) Invoke Device application checks and provides compatibility function and parameters Invoke SM_SetDeviceCom(new supported RID, new parameter list) Invoke SM_SetDeviceIdent(new supported DID, parameter list)
SM_CompStartup_6	In SM_CompatStartup the communication and identification data are reread and verified by the Master SM. The state is left on receiving a DL_Mode(INACTIVE) or a DL_Mode(PREOPERATE) indication.
SM_Preoperate_7	During SM_Preoperate the SerialNumber can be read and verified by the Master SM, as well as Data Storage and Device parameterization may be executed. The state is left on receiving a DL_Mode(INACTIVE), a DL_Mode(STARTUP) or a DL_Mode(OPERATE) indication.
SM_Operate_8	During SM_Operate the cyclic Process Data exchange and acyclic On-request Data transfer are active. The state is left on receiving a DL_Mode(INACTIVE) or a DL_Mode(STARTUP) indication.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	The Device is switched to the configured SIO mode by receiving the trigger SM_SetDeviceMode.req(SIO). Invoke PL_SetMode(DI|DO|INACTIVE) Invoke SM_DeviceMode(SIO)
T2	1	2	The Device is switched to the communication mode by receiving the trigger DL_Mode.ind(ESTABCOM). Invoke PL_SetMode(COMx) Invoke SM_DeviceMode(ESTABCOM)
T3	2,3,4,5,6, 7,8	0	The Device is switched to SM_Idle mode by receiving the trigger DL_Mode.ind(INACTIVE) . Invoke PL_SetMode(INACTIVE) Invoke SM_DeviceMode(IDLE)
T4	2	3	The Device application receives an indication on the baud rate with which the communication has been established in the DL triggered by DL_Mode.ind(COMx). Invoke SM_DeviceMode(COMx)
T5	3	4	The Device identification phase is entered by receiving the trigger DL_Write.ind(MCmd_MASTERIDENT). Invoke SM_DeviceMode(IDENTSTARTUP)
T6	4	5	The Device identity check phase is entered by receiving the trigger DL_Write.ind(MCmd_DEVICEIDENT). Invoke SM_DeviceMode(IDENTCHANGE)
T7	5	6	The Device compatibility startup phase is entered by receiving the trigger DL_Read.ind(Direct Parameter page 1, address 0x02 = "MinCycleTime").
T8	6	7	The Device's preoperate phase is entered by receiving the trigger DL_Mode.ind(PREOPERATE).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
			Invoke SM_DeviceMode(PREOPERATE)
T9	7	8	The Device's operate phase is entered by receiving the trigger DL_Mode.ind(OPERATE). Invoke SM_DeviceMode(OPERATE)
T10	4	7	The Device's preoperate phase is entered by receiving the trigger DL_Mode.ind(PREOPERATE). Invoke SM_DeviceMode(PREOPERATE)
T11	3	8	The Device's operate phase is entered by receiving the trigger DL_Mode.ind(OPERATE). Invoke SM_DeviceMode(OPERATE)
T12	7	3	The Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP)
T13	8	3	The Device's communication startup phase is entered by receiving the trigger DL_Mode.ind(STARTUP). Invoke SM_DeviceMode(STARTUP)

INTERNAL ITEMS	TYPE	DEFINITION
COMx	Variable	Any of COM1, COM2, or COM3 transmission rates
DL_Write_MCmd_xxx	Service	DL Service writes MasterCommands (xxx = values out of Table B.2)

Figure 80 shows a typical sequence chart for the SM communication startup of a Device matching the Master port configuration settings (regular startup).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

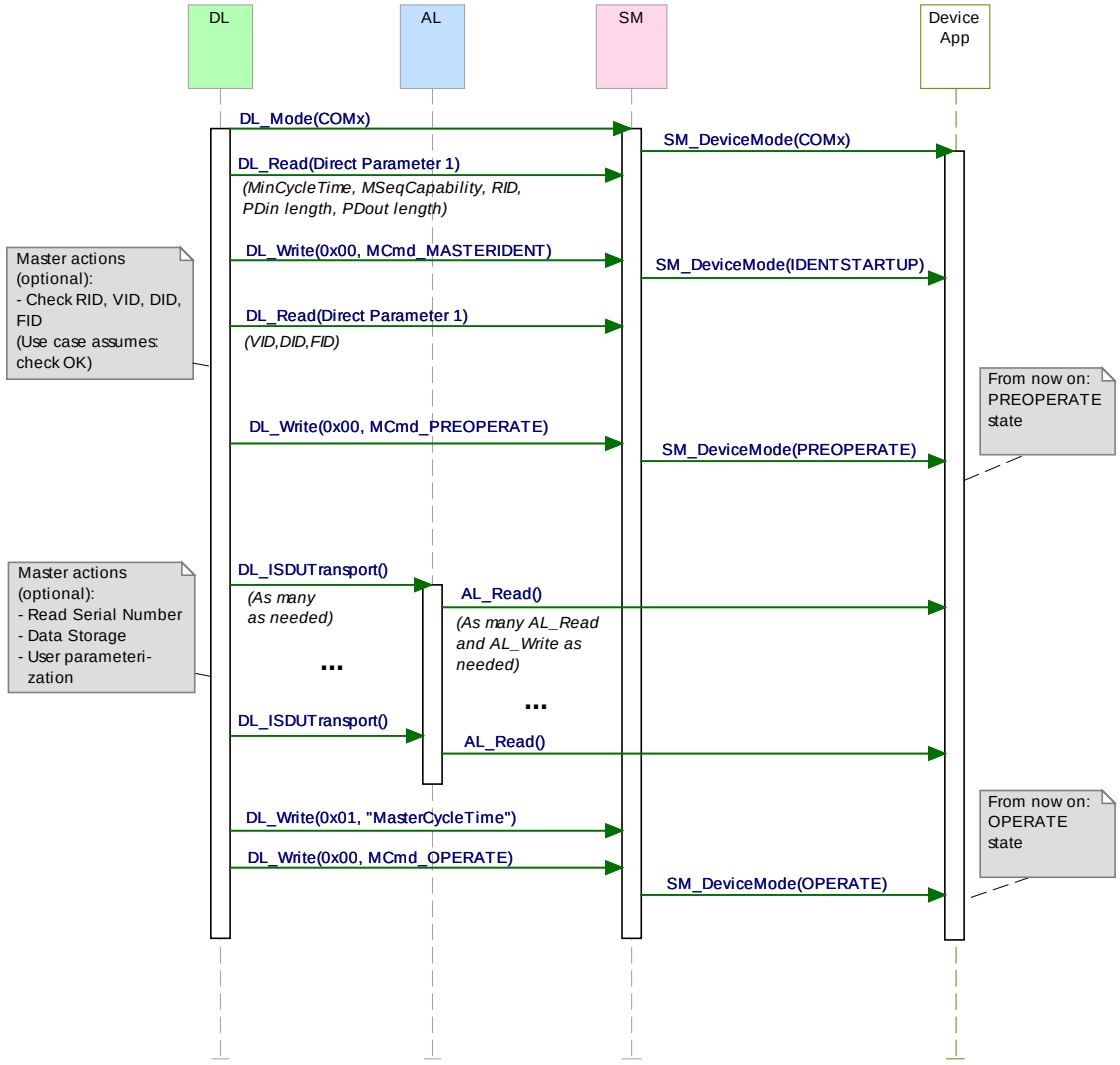


Figure 80 – Sequence chart of a regular Device startup

Figure 81 shows a typical sequence chart for the SM communication startup of a Device not matching the Master port configuration settings (compatibility mode). In this mode, the Master tries to overwrite the Device's identification parameters to achieve a compatible and a workable mode.

The sequence chart in Figure 81 shows only the actions until the PREOPERATE state. The remaining actions until the OPERATE state can be taken from Figure 80.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

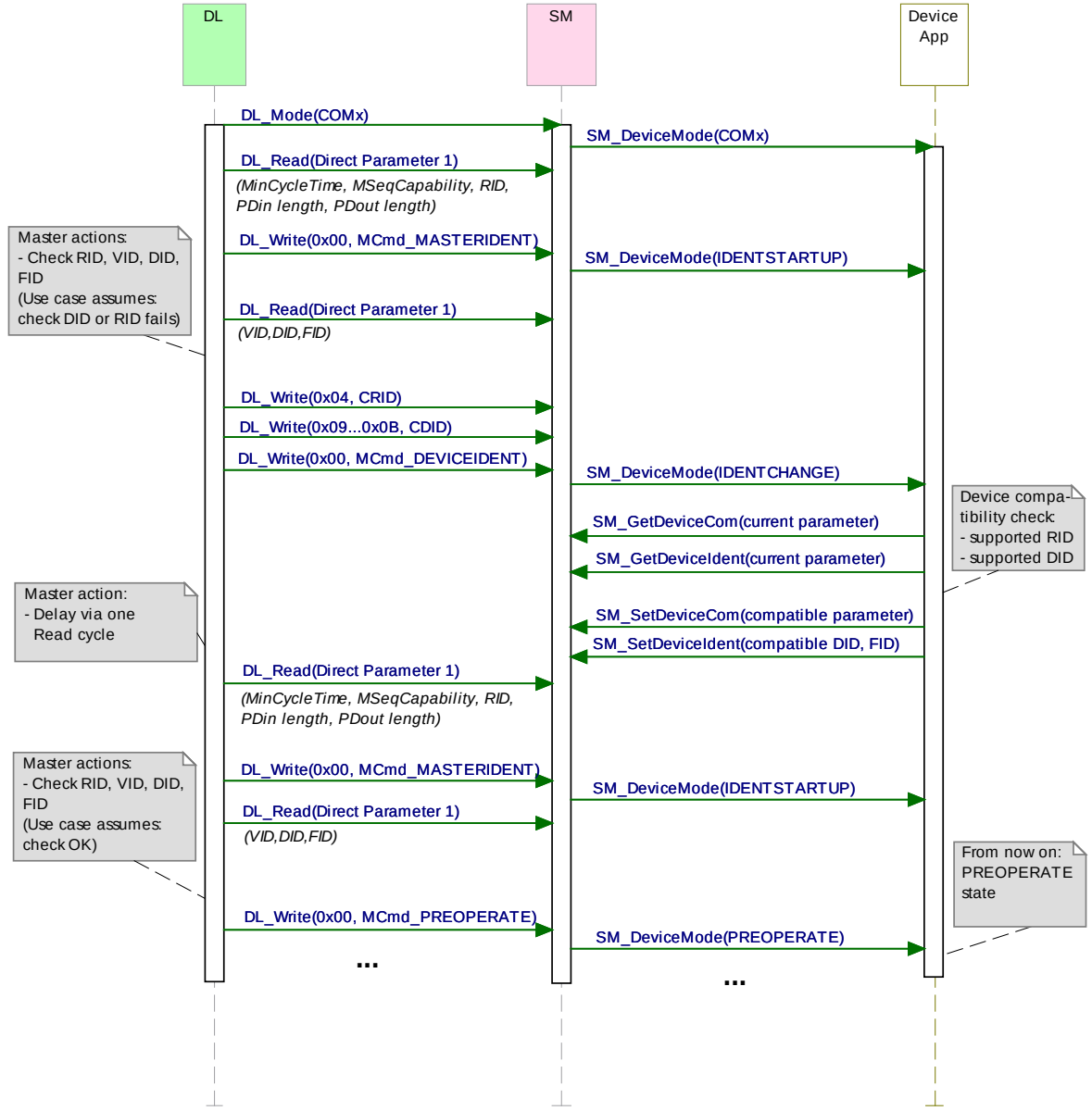


Figure 81 – Sequence chart of a Device startup in compatibility mode

Figure 82 shows a typical sequence chart for the SM communication startup of a Device not matching the Master port configuration settings. The system management of the Master tries to reconfigure the Device with alternative Device identification parameters (compatibility mode). In this use case, the alternative parameters are assumed to be incompatible.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

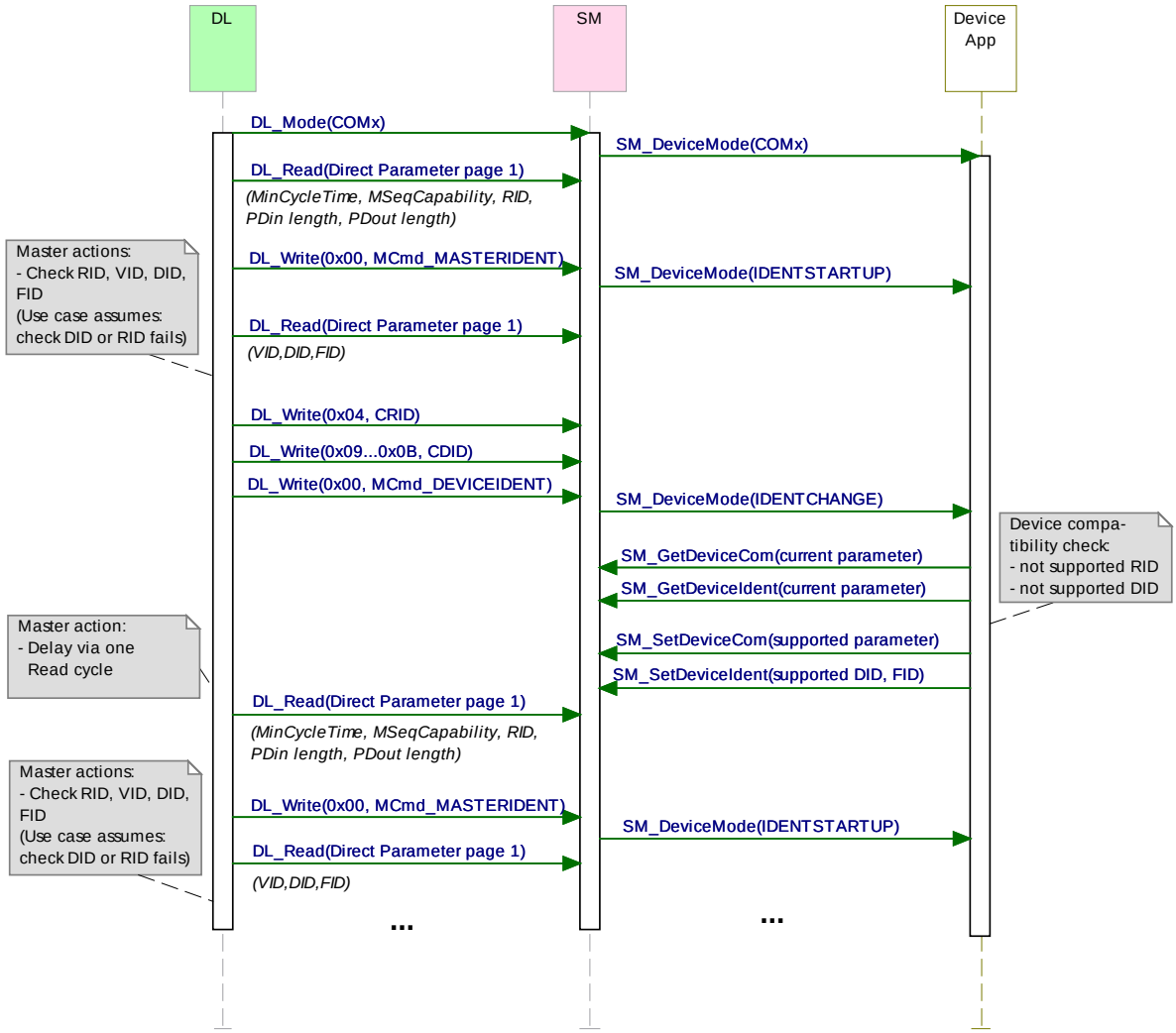


Figure 82 – Sequence chart of a Device startup when compatibility fails

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10 Device

10.1 Overview

Figure 83 provides an overview of the complete structure and services of a Device.

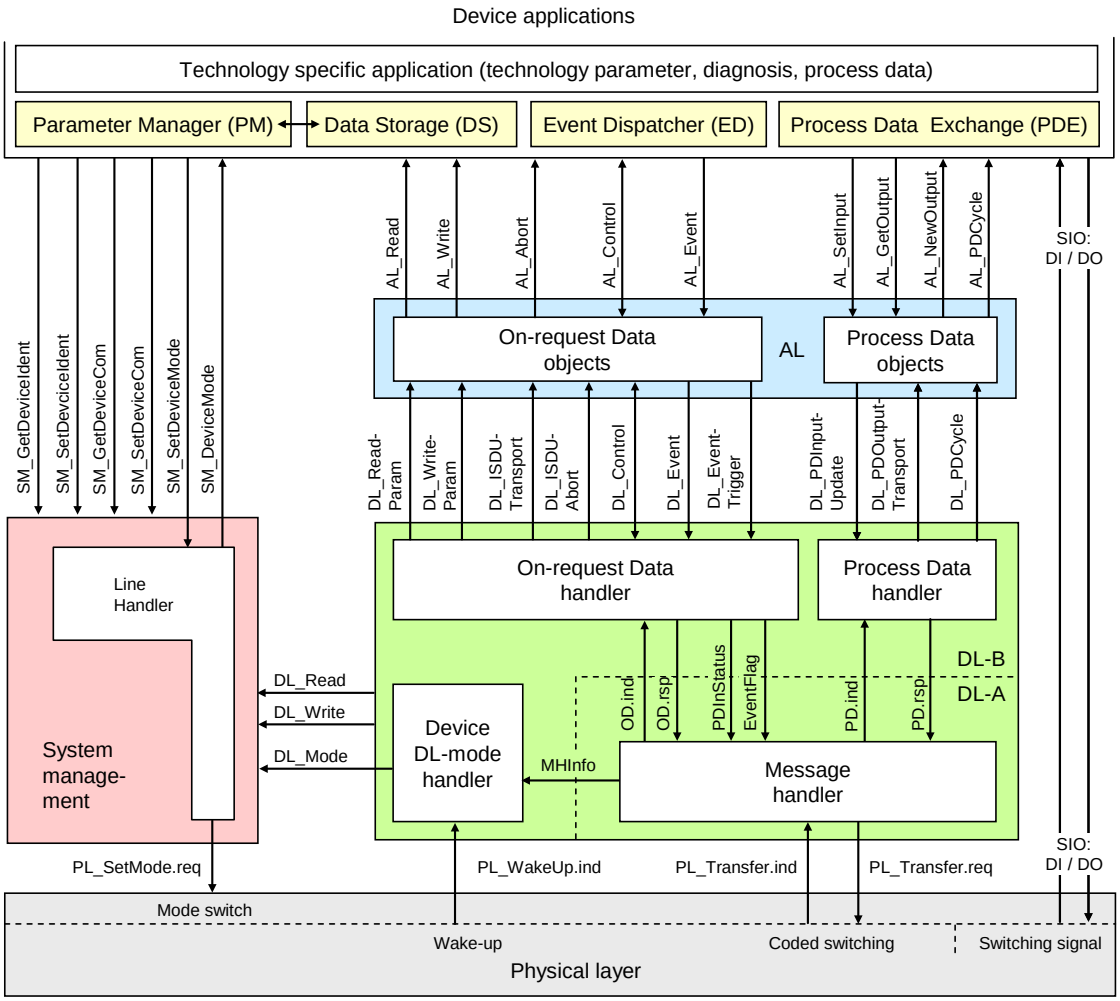


Figure 83 – Structure and services of a Device

The Device applications comprise first the technology specific application consisting of the transducer with its technology parameters, its diagnosis information, and its Process Data. The common Device applications comprise:

- Parameter Manager (PM), dealing with compatibility and correctness checking of complete sets of technology (vendor) specific and common system parameters (see 10.3);
- Data Storage (DS) mechanism, which optionally uploads or downloads parameters to the Master (see 10.4);
- Event Dispatcher (ED), supervising states and conveying diagnosis information such as notifications, warnings, errors, and Device requests as peripheral initiatives (see 10.5);
- Process Data Exchange (PDE) unit, conditioning the data structures for transmission in case of a sensor or preparing the received data structures for signal generation. It also controls the operational states to ensure the validity of Process Data (see 10.2).

These Device applications provide standard methods/functions and parameters common to all Devices, and Device specific functions and parameters, all specified within Clause 10.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.2 Process Data Exchange (PDE)

The Process Data Exchange unit cyclically transmits and receives Process Data without interference from the On-request Data (parameters, commands, and Events).

An actuator (output Process Data) shall observe the cyclic transmission and enter a default appropriate state, for example keep last value, stop, or de-energize, whenever the data transmission is interrupted (see 7.3.3.5 and 10.7.3). The actuator shall wait on the MasterCommand "ProcessDataOutputOperate" (see Table B.2, output Process Data "valid") prior to regular operation after restart in case of an interruption.

Within cyclic data exchange, an actuator (output Process Data) receives a Master-Command "DeviceOperate", whenever the output Process Data are invalid and a Master-Command "ProcessDataOutputOperate", whenever they become valid again (see Table B.2).

There is no need for a sensor Device (input Process Data) to monitor the cyclic data exchange. However, if the Device is not able to guarantee valid Process Data, the PD status "Process Data invalid" (see A.1.5) shall be signaled to the Master application.

10.3 Parameter Manager (PM)

10.3.1 General

A Device can be parameterized via two basic methods using the Direct Parameters or the Index memory space accessible with the help of ISDUs (see Figure 5).

Mandatory for all Devices are the so-called Direct Parameters in page 1. This page 1 contains common communication and identification parameters (see B.1).

Direct Parameter page 2 optionally offers space for a maximum of 16 octets of technology (vendor) specific parameters for Devices requiring not more than this limited number and with small system footprint (ISDU communication not implemented, easier fieldbus handling possible but with less comfort). Access to the Direct Parameter page 2 is performed via AL_Read and AL_Write (see 10.7.5).

The transmission of parameters to and from the spacious Index memory can be performed in two ways: single parameter by single parameter or as a block of parameters. Single parameter transmission as specified in 10.3.4 is secured via several checks and confirmation of the transmitted parameter. A negative acknowledgement contains an appropriate error description and the parameter is not activated. Block parameter transmission as specified in 10.3.5 defers parameter consistency checking and activation until after the complete transmission. The Device performs the checks upon reception of a special command and returns a confirmation or a negative acknowledgement with an appropriate error description. In this case the transmitted parameters shall be rejected and a roll back to the previous parameter set shall be performed to ensure proper functionality of the Device.

10.3.2 Parameter manager state machine

The Device can be parameterized using ISDU mechanisms whenever the PM is active. The main functions of the PM are the transmission of parameters to the Master ("Upload"), to the Device ("Download"), and the consistency and validity checking within the Device ("ValidityCheck") as demonstrated in Figure 84.

The PM is driven by command messages of the Master (see Table B.9). For example the guard [UploadStart] corresponds to the reception of the SystemCommand "ParamUploadStart" and [UploadEnd] to the reception of the SystemCommand "ParamUploadEnd".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE 1 Following a communication interruption, the Master system management uses the service SM_DeviceMode with the variable "INACTIVE" to stop the upload process and to return to the "IDLE" state.

Any new "ParamUploadStart" or "ParamDownloadStart" while another sequence is pending, for example due to an unexpected shut-down of a vendor parameterization tool, will abort the pending sequence. The corresponding parameter changes will be discarded.

NOTE 2 A PLC user program and a parameterization tool can conflict (multiple access), for example if during commissioning, the user did not disable accesses from the PLC program while changing parameters via the tool.

The parameter manager mechanism in a Device is always active and the DS_ParUpload.req in transition T4 is used to trigger the Data Storage (DS) mechanism in 10.4.2.

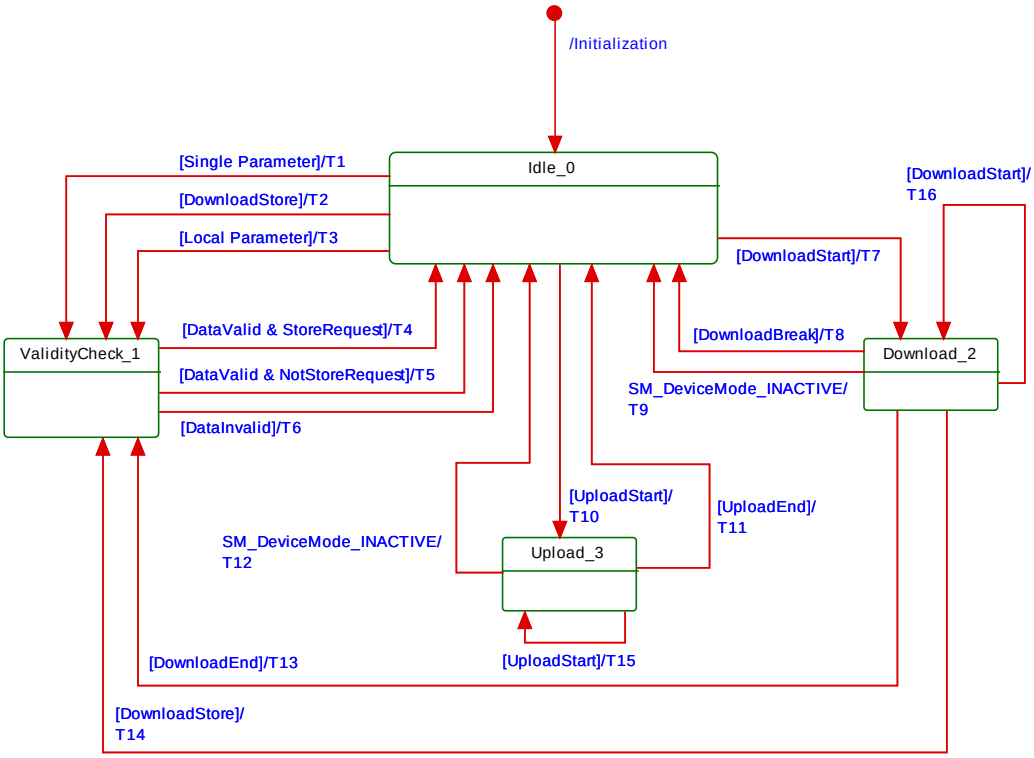


Figure 84 – The Parameter Manager (PM) state machine

Table 94 shows the state transition tables of the Device Parameter Manager (PM) state machine.

Table 94 – State transition tables of the PM state machine

STATE NAME	STATE DESCRIPTION
Idle_0	Waiting on parameter transmission
ValidityCheck_1	Check of consistency and validity of current parameter set.
Download_2	Parameter download active; local parameterization locked (e.g. teach-in)
Upload_3	Parameter upload active; parameterization globally locked; all write accesses for parameter changes via tools shall be rejected (ISDU ErrorCode "Service temporarily not available – Device control")

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	0	1	Set "StoreRequest" (= TRUE)
T3	0	1	Set "StoreRequest" (= TRUE)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T4	1	0	Mark parameter set as valid; invoke DS_ParUpload.req to DS; enable positive acknowledge of transmission; reset "StoreRequest" (= FALSE)
T5	1	0	Mark parameter set as valid; enable positive acknowledge of transmission
T6	1	0	Mark parameter set as invalid; enable negative acknowledge of transmission; reset "StoreRequest" (= FALSE); discard parameter buffer
T7	0	2	Lock local parameter access
T8	2	0	Unlock local parameter access; discard parameter buffer
T9	2	0	Unlock local parameter access; discard parameter buffer
T10	0	3	Lock local parameter access
T11	3	0	Unlock local parameter access
T12	3	0	Unlock local parameter access
T13	2	1	Unlock local parameter access
T14	2	1	Unlock local parameter access; set "StoreRequest" (= TRUE)
T15	3	3	Lock local parameter access
T16	2	2	Discard parameter buffer, so that a possible second start will not be blocked.

INTERNAL ITEMS	TYPE	DEFINITION
DownloadStore	Bool	SystemCommand "ParamDownloadStore" received, see Table B.9
DataValid	Bool	Positive result of conformity and validity checking
DataInvalid	Bool	Negative result of conformity and validity checking
DownloadStart	Bool	SystemCommand "ParamDownloadStart" received, see Table B.9
DownloadBreak	Bool	SystemCommand "ParamBreak" or "ParamUploadStart" received
DownloadEnd	Bool	SystemCommand "ParamDownloadEnd" received, see Table B.9
StoreRequest	Bool	Flag for a requested Data Storage sequence, i.e. SystemCommand "ParamDownloadStore" received (= TRUE)
NotStoreRequest	Bool	Inverted value of StoreRequest
UploadStart	Bool	SystemCommand "ParamUploadStart" received, see Table B.9
UploadEnd	Bool	SystemCommand "ParamUploadEnd" received, see Table B.9
Single Parameter	Bool	In case of "single parameter" as specified in 10.3.4
Local Parameter	Bool	In case of "local parameter" as specified in 10.3.3

The Parameter Manager (PM) supports handling of "single parameter" (Index and Subindex) transfers as well as "block parameter" transmission (entire parameter set).

10.3.3 Dynamic parameter

Parameters accessible through SDCI read or write services may also be changed via on-board control elements (for example teach-in button) or the human machine interface of a Device. These changes shall undergo the same validity checks as a single parameter access. Thus, in case of a positive result "DataValid" in Figure 84, the "StoreRequest" flag shall be applied in order to achieve Data Storage consistency. In case of a negative result "InvalidData", the previous values of the corresponding parameters shall be restored ("roll back"). In addition, a Device specific indication on the human machine interface is recommended as a positive or negative feedback to the user.

It is recommended to avoid concurrent access to a parameter via local control elements and SDCI write services at the same point in time.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.3.4 Single parameter

Sample sequence charts for valid and invalid single parameter changes are specified in Figure 85.

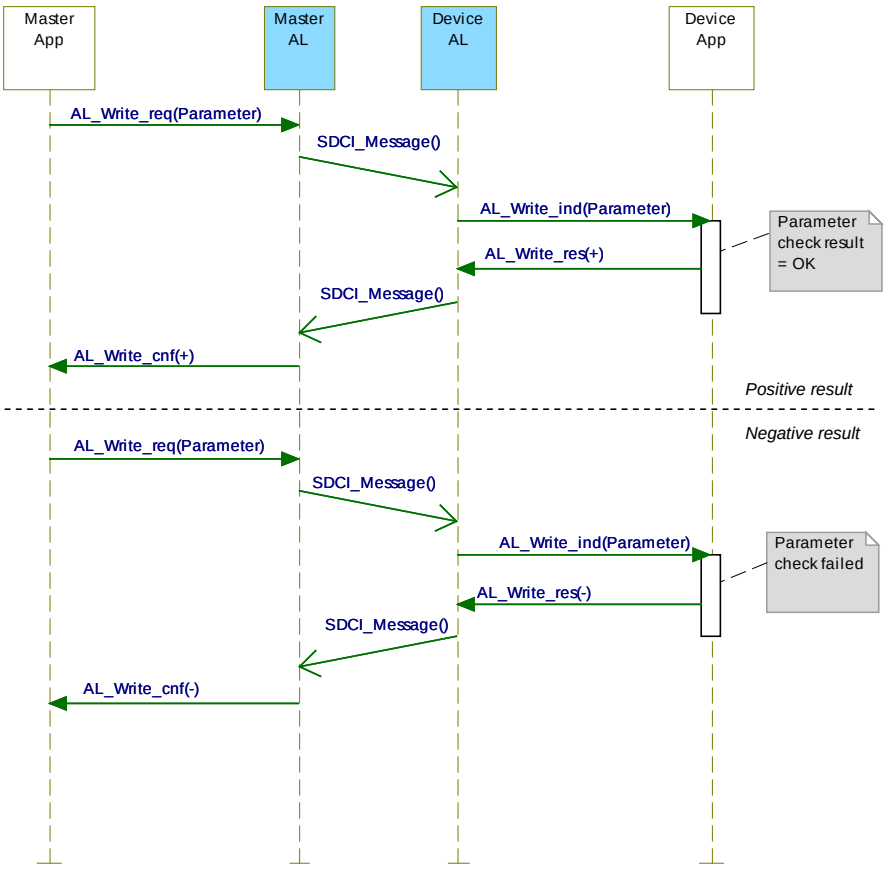


Figure 85 – Positive and negative parameter checking result

If single parameterization is performed via ISDU objects, the Device shall check the access, structure, consistency and validity (see Table 95) of the transmitted data within the context of the entire parameter set and return the result in the confirmation. The negative confirmation carries one of the error indications of Table C.2.

Table 95 – Definitions of parameter checks

Parameter check	Definition	Error indication
Access	Check for valid access rights for this Index / Subindex, independent from data content (Index / Subindex permanent or temporarily unavailable; write access on read only Index)	See C.2.3 to C.2.8
Consistency	Check for valid data content of the entire parameter set, testing for interference or correlations between parameters	See C.2.16 and C.2.17
Structure	Check for valid data structure like data size, only complete data structures can be written, for example 2 octets to an UInteger16 data type	See C.2.12 and C.2.13
Validity	Check for valid data content of single parameters, testing for data limits	See C.2.9 to C.2.11, C.2.14, C.2.15

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.3.5 Block parameter

User applications such as function blocks within PLCs and parameterization tool software can use start and end commands to indicate the begin and end of a block parameter transmission. For the duration of the block parameter transmission the Device application shall inhibit all the parameter changes originating from other sources, for example local parameterization, teach-in, etc.

A sample sequence chart for valid block parameter changes with an optional Data Storage request is demonstrated in Figure 86.

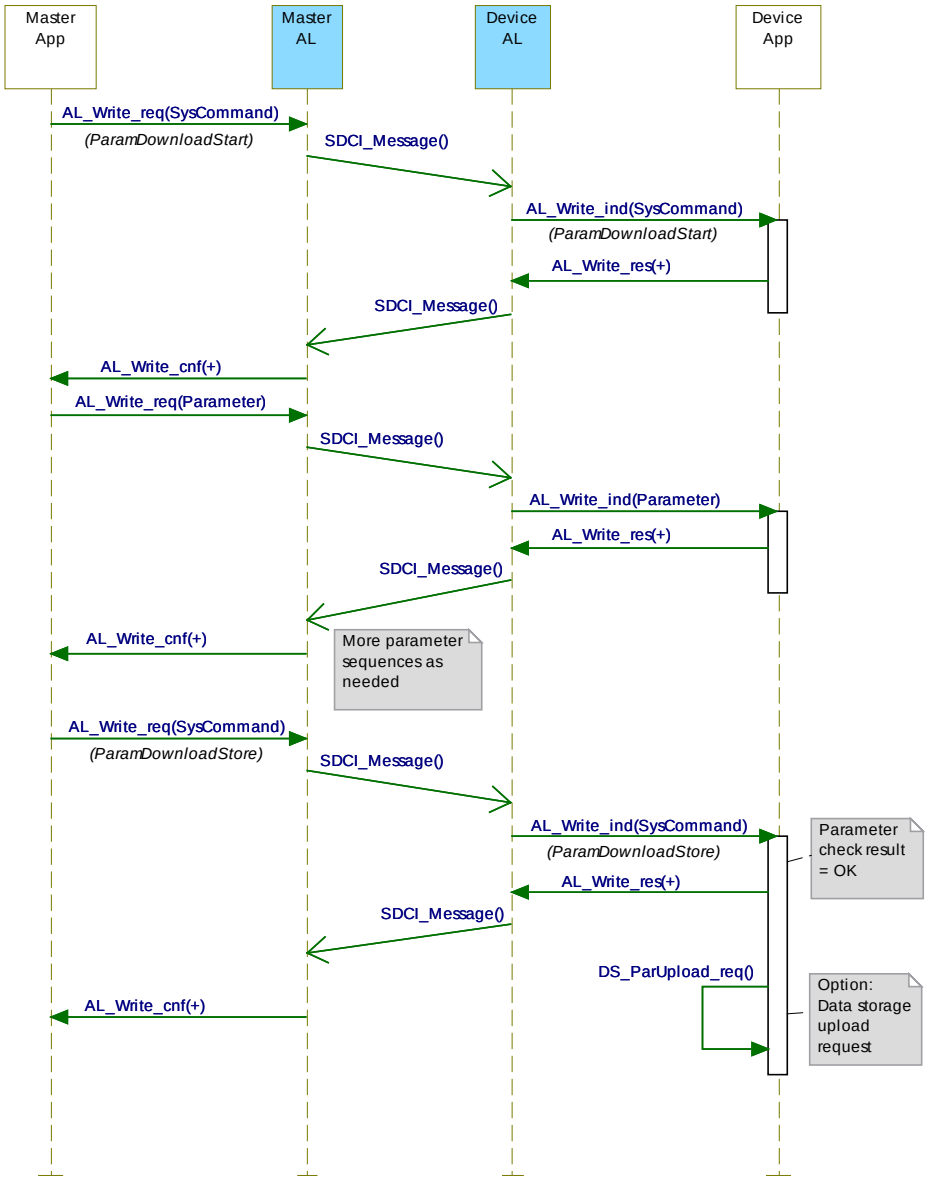


Figure 86 – Positive block parameter download with Data Storage request

A sample sequence chart for invalid block parameter changes is demonstrated in Figure 87.

The "ParamDownloadStart" command (see Table B.9) indicates the beginning of the block parameter transmission in download direction (from user application to the Device). The SystemCommand "ParamDownloadEnd" or "ParamDownloadStore" terminates this sequence. Both functions are similar. However, in addition the SystemCommand "ParamDownloadStore"

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

causes the Data Storage (DS) mechanism to upload the parameter set through the DS_UPLOAD_REQ Event (see 10.4.2).

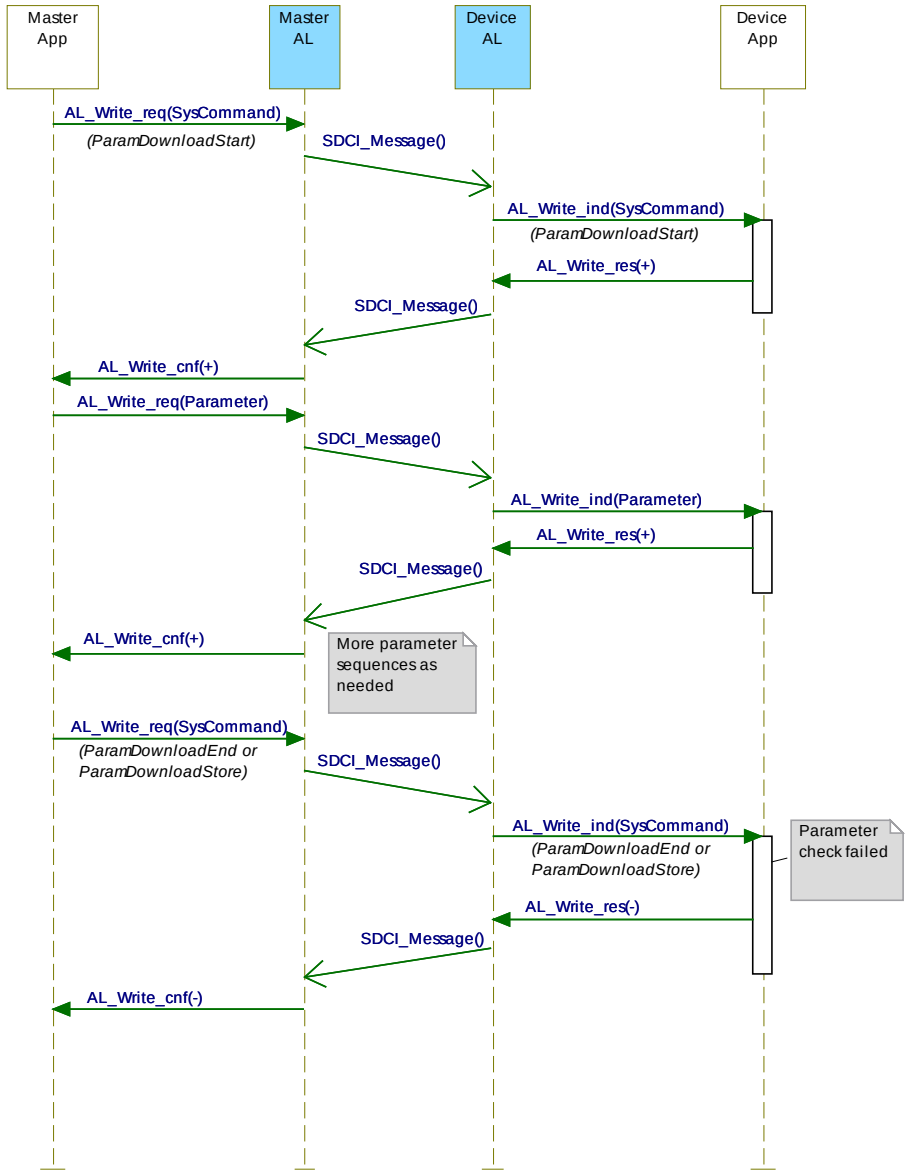


Figure 87 – Negative block parameter download

During block parameter download the consistency checking for single transferred parameters shall be disabled and the parameters are not activated. With the "ParamDownloadEnd" command, the Device checks the entire parameter set and indicates the result to the originator of the block parameter transmission within the ISDU acknowledgement in return to the command.

During the block parameter download the access and structure checks are always performed (see Table 95). Optionally, validity checks may also be performed. The parameter manager shall not exit from the block transfer mode in case of invalid accesses or structure violations.

In case of an invalid parameter set the changed parameters shall be discarded and a rollback to the previous parameter set shall be performed. The corresponding negative confirmation shall contain one of the error indications from Table C.2. With a negative confirmation of the SystemCommand "ParamDownloadStore", the Data Storage upload request is omitted.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The "ParamUploadStart" command (see Table B.9) indicates the beginning of the block parameter transmission in upload direction (from the Device to the user application). The SystemCommand "ParamUploadEnd" terminates this sequence and indicates the end of transmission.

A block parameter transmission is aborted if the parameter manager receives a SystemCommand "ParamBreak". In this case the block transmission quits without any changes in parameter settings.

10.3.6 Concurrent parameterization access

There is no mechanism to secure parameter consistency within the Device in case of concurrent accesses from different user applications above Master level. This shall be ensured or blocked on user level.

10.3.7 Command handling

Application commands such as teach-in or restore factory settings are conveyed in form of parameters. An application command is confirmed with a positive service response – AL_Write.res(+). A negative service response – AL_Write.res(-) – shall indicate the failed execution of the application command. In both cases the ISDU timeout limit shall be considered (see Table 97).

10.4 Data Storage (DS)

10.4.1 General

The Data Storage (DS) mechanism enables the consistent and up-to-date buffering of the Device parameters on upper levels like PLC programs or fieldbus parameter server. Data Storage between Masters and Devices is specified within this standard, whereas the adjacent upper data storage mechanisms depend on the individual fieldbus or system. The Device holds a standardized set of objects providing information about parameters for Data Storage such as memory size requirements, control and state information of the Data Storage mechanism (see Table B.10). Revisions of Data Storage parameter sets are identified via a Parameter Checksum.

The implementation of the DS mechanism specified in this standard is highly recommended for Devices. If this mechanism is not supported it is the responsibility of the Device vendor to describe how parameterization of a Device after replacement can be ensured in a system conform manner without tools.

10.4.2 Data Storage state machine

Any changed set of valid parameters leads to a new Data Storage upload. The upload is initiated by the Device by raising a "DS_UPLOAD_REQ" Event (see Table D.2). The Device shall store the internal state "Data Storage Upload" in non-volatile memory (see Table B.10, State Property), until it receives a Data Storage command "DS_UploadEnd" or "DS_DownloadEnd".

The Device shall generate an Event "DS_UPLOAD_REQ" (see Table D.2) only if the parameter set is valid and

- parameters assigned for Data Storage have been changed locally on the Device (for example teach-in, human machine interface, etc.), or
- the Device receives a SystemCommand "ParamDownloadStore".

With this Event information the Data Storage mechanism of the Master is triggered and initiates a Data Storage upload sequence.

The state machine in Figure 88 specifies the Device Data Storage mechanism.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

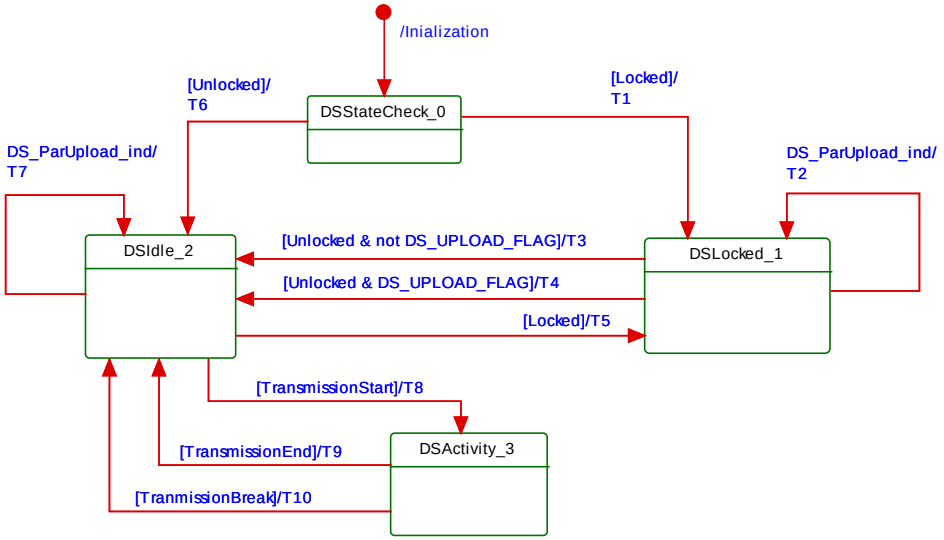


Figure 88 – The Data Storage (DS) state machine

Table 96 shows the state transition tables of the Device Data Storage (DS) state machine. See Table B.10 for details on Data Storage Index assignments.

Table 96 – State transition table of the Data Storage state machine

STATE NAME	STATE DESCRIPTION
DSStateCheck_0	Check activation state after initialization.
DSLocked_1	Waiting on Data Storage state machine to become unlocked.
DSIdle_2	Waiting on Data Storage activities.
DSActivity_3	Provide parameter set; local parameterization locked.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	Set State_Property = "Data Storage access locked"
T2	1	1	Set DS_UPLOAD_FLAG = TRUE
T3	1	2	Set State_Property = "Inactive"
T4	1	2	Invoke AL_EVENT.req (EventCode: DS_UPLOAD_REQ), Set State_Property = "Inactive"
T5	2	1	Set State_Property = "Data Storage access locked"
T6	0	2	Set State_Property = "Inactive"
T7	2	2	Set DS_UPLOAD_FLAG = TRUE, invoke AL_EVENT.req (EventCode: DS_UPLOAD_REQ)
T8	2	3	Lock local parameter access, set State_Property = "Upload" or "Download"
T9	3	2	Set DS_UPLOAD_FLAG = FALSE, unlock local parameter access, Set State_Property = "Inactive"
T10	3	2	Unlock local parameter access, Set State_Property = "Inactive"

INTERNAL ITEMS	TYPE	DEFINITION
Unlocked	Bool	Data Storage unlocked, see B.2.4
Locked	Bool	Data Storage locked, see B.2.4
DS_ParUpload.ind	Service	Device internal service between PM and DS (see Figure 84)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
TransmissionStart	Bool	DS_Command "DS_UploadStart" or "DS_DownloadStart" has been invoked
TransmissionEnd	Bool	DS_Command "DS_UploadEnd" or "DS_DownloadEnd" has been invoked
TransmissionBreak	Bool	SM_MODE_INACTIVE or DS_Command "DS_Break" received

The truncated sequence chart in Figure 89 demonstrates the important communication sequences after the parameterization.

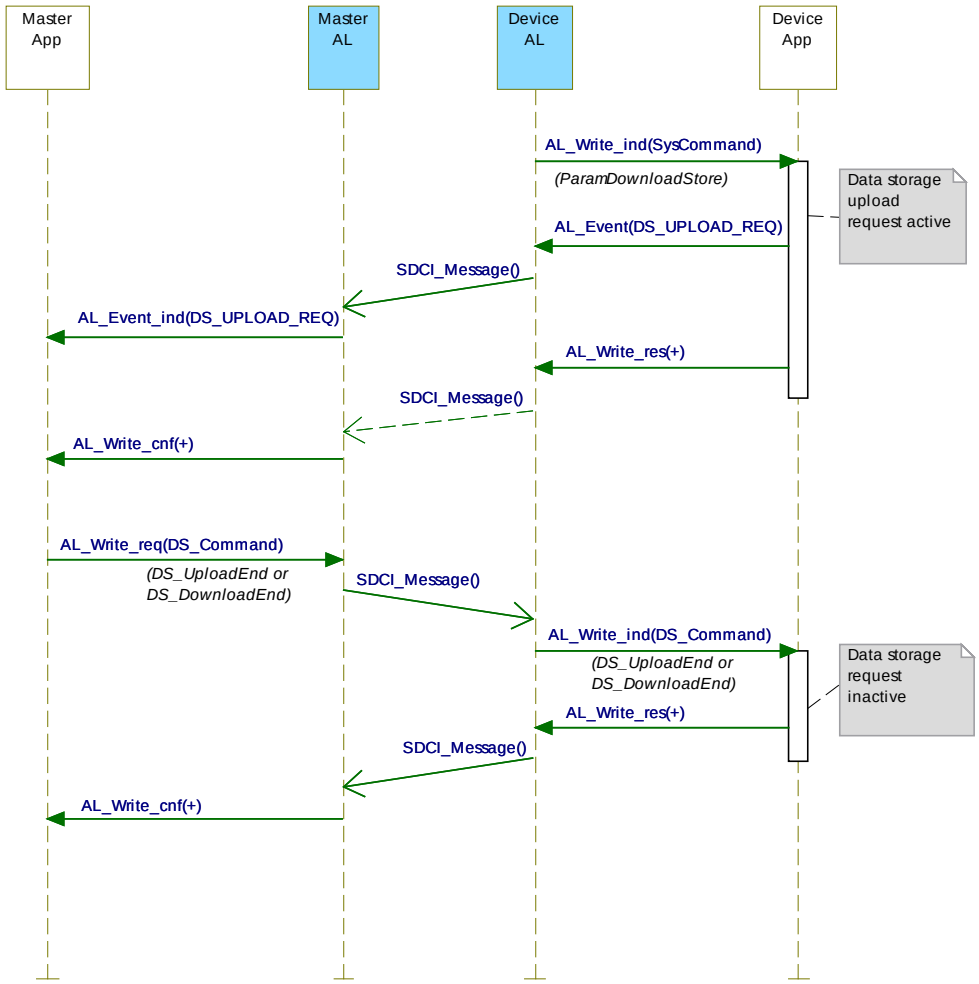


Figure 89 – Data Storage request message sequence

10.4.3 DS configuration

The Data Storage mechanism inside the Device may be disabled via the Master, for example by a tool or a PLC program. See B.2.4 for further details.

This is recommended during commissioning or system tests to avoid intensive communication.

10.4.4 DS memory space

To handle the requested data amount for Data Storage under any circumstances, the requested amount of indices to be saved and the required total memory space are given in the Data Storage Size parameter, see Table B.10. The required total memory space (including

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

the structural information shall not exceed 2 048 octets (see Annex F). The Data Storage mechanism of the Master shall be able to support this amount of memory per port.

10.4.5 DS Index_List

The Device is the "owner" of the DS Index_List (see Table B.10). Its purpose is to provide all the necessary information for a Device replacement. The DS Index_List shall be fixed for any specific DeviceID. Otherwise the data integrity between Master and Device cannot be guaranteed. The Index List shall contain the termination marker (see Table B.10), if the Device does not support Data Storage (see 10.4.1). The required storage size shall be 0 in this case.

10.4.6 DS parameter availability

All indices listed in the Index List shall be readable and writeable between the SystemCommands "DS_UploadStart" or "DS_DownloadStart" and "DS_UploadEnd" or "DS_DownloadEnd" (see Table B.10). If one of the Indices is rejected by the Device, the Data Storage Master will abort the up- or download with a SystemCommand "DS_Break". In this case no retries of the Data Storage sequence will be performed.

10.4.7 DS without ISDU

The support of ISDU transmission in a Device is a precondition for the Data Storage of parameters. Parameters in Direct Parameter page 2 cannot be saved and restored by the Data Storage mechanism.

10.4.8 DS parameter change indication

The Parameter_Checksum specified in Table B.10 is used as an indicator for changes in a parameter set. This standard does not require a specific mechanism for detecting parameter changes. A set of recommended methods is provided in the informative Annex J.

10.5 Event Dispatcher (ED)

Any of the Device applications can generate predefined system status information when SDCI operations fail or technology specific information (diagnosis) as a result from technology specific diagnostic methods occur. The Event Dispatcher turns this information into an Event according to the definitions in A.6. The Event consists of an EventQualifier indicating the properties of an incident and an EventCode ID representing a description of this incident together with possible remedial measures. Table D.1 comprises a list of predefined IDs and descriptions for application oriented incidents. Ranges of IDs are reserved for profile specific and vendor specific incidents. Table D.2 comprises a list of predefined IDs for SDCI specific incidents.

Events are classified in "Errors", "Warnings", and "Notifications". See 10.9.2 for these classifications and see 11.5 for how the Master is controlling and processing these Events.

All Events provided at one point in time are acknowledged with one single command. Therefore the Event acknowledgement may be delayed by the slowest acknowledgement from upper system levels.

10.6 Device features

10.6.1 General

The following Device features are defined to a certain degree in order to achieve a common behavior. They are accessible via standardized or Device specific methods or parameters. The availability of these features is defined in the IODD of a Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.6.2 Device backward compatibility

This feature enables a Device to play the role of a previous Device revision. In the start-up phase the Master system management overwrites the Device's inherent DeviceID (DID) with the requested former DeviceID. The Device's technology application shall switch to the former functional sets or subsets assigned to this DeviceID. Device backward compatibility support is optional for a Device.

As a Device can provide backward compatibility to previous DeviceIDs (DID), these compatible Devices shall support all parameters and communication capabilities of the previous Device ID. Thus, the Device is permitted to change any communication or identification parameter in this case.

10.6.3 Protocol revision compatibility

This feature enables a Device to adjust its protocol layers to a previous SDCI protocol version such as for example to the legacy protocol version of a legacy Master or in the future from version V(x) to version V(x-n). In the start-up phase the Master system management can overwrite the Device's inherent protocol RevisionID (RID) in case of discrepancy with the RevisionID supported by the Master. A legacy Master does not write the MasterCommand "MasterIdent" (see Table B.2) and thus the Device can adjust to the legacy protocol (V1.0). Revision compatibility support is optional for a Device.

10.6.4 Factory settings

This feature enables a Device to restore parameters to the original delivery status. The Data Storage flag and other dynamic parameters such as "Error Count" (see B.2.17), "Device Status" (see B.2.18), and "Detailed Device Status" (see B.2.19) shall be reset when this feature is applied. This does not include vendor specific parameters such as for example counters of operating hours.

NOTE In this case an existing stored parameter set within the Master will be automatically downloaded into the Device after its start-up.

It is the vendor's responsibility to guarantee the correct function under any circumstances. The reset is triggered by the reception of the SystemCommand "Restore factory settings" (see Table B.9). Reset to factory settings is optional for a Device.

10.6.5 Application reset

This feature enables a Device to reset the technology specific application. It is especially useful whenever a technology specific application has to be set to a predefined operational state without communication interruption and a shut-down cycle. The reset is triggered by the reception of a SystemCommand "Application reset" (see Table B.9). Reset of the technology specific application is optional for a Device.

10.6.6 Device reset

This feature enables a Device to perform a "warm start". It is especially useful whenever a Device has to be reset to an initial state such as power-on. In this case communication will be interrupted. The warm start is triggered by the reception of a SystemCommand "Device reset" (see Table B.9). Warm start is optional for a Device.

10.6.7 Visual SDCI indication

This feature indicates the operational state of the Device's SDCI interface. The indication of the SDCI mode is specified in 10.9.3. Indication of the SIO mode is vendor specific and not covered by this definition. The function is triggered by the indication of the system management (within all states except SM_Idle and SM_SIO in Figure 79). SDCI indication is optional for a Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.6.8 Parameter access locking

This feature enables a Device to globally lock or unlock write access to all writeable Device parameters accessible via the SDCI interface (see B.2.4). The locking is triggered by the reception of a system parameter "Device Access Locks" (see Table B.8). The support for these functions is optional for a Device.

10.6.9 Data Storage locking

Setting this lock will cause the "State_Property" in Table B.10 to switch to "Data Storage locked" and the Device not to send a DS_UPLOAD_REQ Event. The support for this function is mandatory for a Device if the Data Storage mechanism is implemented.

10.6.10 Device parameter locking

Setting this lock will disable overwriting Device parameters via on-board control or adjustment elements such as teach-in buttons (see B.2.4). The support of this function is optional for a Device.

10.6.11 Device user interface locking

Setting this lock will disable the operation of on-board human machine interface displays and adjustment elements such as teach-in buttons on a Device (see B.2.4). The support for this function is optional for a Device.

10.6.12 Offset time

The offset time t_{offset} is a parameter to be configured by the user (see B.2.22). It determines the beginning of the Device's technology data processing in respect to the start of the M-sequence cycle, that means the beginning of the Master (port) message.

The offset enables:

- data processing of a Device to be synchronized with the Master (port) cycle within certain limits;
- data processing of multiple Devices on different Master ports to be synchronized with one another;
- data processing of multiple Devices on different Master ports to run with a defined offset.

Figure 90 demonstrates the timing of messages in respect to the data processing in Devices.

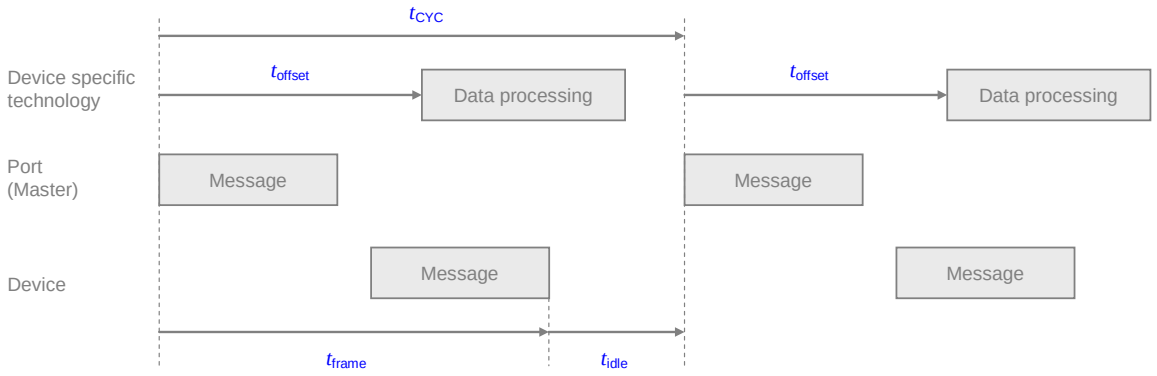


Figure 90 – Cycle timing

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The offset time defines a trigger relative to the start of an M-sequence cycle. The support for this function is optional for a Device.

10.6.13 Data Storage concept

The Data Storage mechanism in a Device allows to automatically save parameters in the Data Storage server of the Master and to restore them upon Event notification. Data consistency is checked in either direction within the Master and Device. Data Storage mainly focuses on configuration parameters of a Device set up during commissioning (see 10.4 and 11.3). The support of this function is optional for a Device.

10.6.14 Block Parameter

The Block Parameter transmission feature in a Device allows transfer of parameter sets from a PLC program without checking the consistency single data object by single data object. The validity and consistency check is performed at the end of the Block Parameter transmission for the entire parameter set. This function mainly focuses on exchange of parameters of a Device to be set up at runtime (see 10.3). The support of this function is optional for a Device.

10.7 Device design rules and constraints

10.7.1 General

In addition to the protocol definitions in form of state, sequence, activity, and timing diagrams some more rules and constraints are required to define the behavior of the Devices. An overview of the major protocol variables scattered all over the standard is concentrated in Table 97 with associated references.

10.7.2 Process Data

The process communication channel transmits the cyclic Process Data without any interference of the On-request Data communication channels. Process Data exchange starts automatically whenever the Device is switched into the OPERATE state via message from the Master.

The format of the transmitted data is Device specific and varies from no data octets up to 32 octets in each communication direction.

Recommendations.

- Data structures should be suitable for use by PLC applications.
- It is highly recommended to comply with the rules in E.3.3 and in [6].

See A.1.5 for details on the indication of valid or invalid Process Data via a PDValid flag within cyclic data exchange.

10.7.3 Communication loss

It is the responsibility of the Device designer to define the appropriate behaviour of the Device in case communication with the Master is lost (transition T10 in Figure 42 handles detection of the communication loss, while 10.2 defines resulting Device actions).

NOTE This is especially important for actuators such as valves or motor management.

10.7.4 Direct Parameter

The Direct Parameter page communication provides no handshake mechanism to ensure proper reception or validity of the transmitted parameters. The Direct Parameter page can only be accessed single octet by single octet (Subindex) or as a whole (16 octets). Therefore,

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

the consistency of parameters larger than 1 octet cannot be guaranteed in case of single octet access.

The parameters from the Direct Parameter page cannot be saved and restored via the Data Storage mechanism.

10.7.5 ISDU communication channel

The ISDU communication channel provides a powerful means for the transmission of parameters and commands (see Clause B.2).

The following rules shall be considered when using this channel (see Figure 6).

- Index 0 is not accessible via the ISDU communication channel. The access is redirected by the Master to the Direct Parameter page 1 using the page communication channel.
- Index 1 is not accessible via the ISDU communication channel. The access is redirected by the Master to the Direct Parameter page 2 using the page communication channel.
- Index 3 cannot be accessed by a PLC application program. The access is limited to the Master application only (Data Storage).
- After reception of an ISDU request from the Master the Device shall respond within 5 000 ms (see Table 97). Any violation causes the Master to abandon the current task.

10.7.6 DeviceID rules related to Device variants

Devices with a certain DeviceID and VendorID shall not deviate in communication and functional behavior. This applies for sensors and actuators. Those Devices may vary for example in

- cable lengths,
- housing materials,
- mounting mechanisms,
- other features, and environmental conditions.

10.7.7 Protocol constants

Table 97 gives an overview of the major protocol constants for Devices.

Table 97 – Overview of the protocol constants for Devices

System variable	References	Values	Definition
ISDU acknowledgement time, for example after a SystemCommand	B.2.2	5 000 ms	Time from reception of an ISDU for example SystemCommand and the beginning of the response message of the Device (see Figure 61)
Maximum number of entries in Index List	B.2.3	70	Each entry comprises an Index and a Subindex. 70 entries results in a total of 210 octets.
Preset values for unused or reserved parameters, for example FunctionID	Annex B	0 (if numbers) 0x00 (if characters)	Engineering shall set all unused parameters to the preset values.
Wake-up procedure	7.3.2.2	See Table 40 and Table 41	Minimum and maximum timings and number of retries
MaxRetry	7.3.3.3	2, see Table 44	Maximum number of retries after communication errors
MinCycleTime	A.3.7 and B.1.3	See Table A.11 and Table B.3	Device defines its minimum cycle time to aquire input or process output data.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

System variable	References	Values	Definition
Usable Index range	Clause B.2	See Table B.8	This version of the standard reserves some areas within the total range of 65 535 Indices.
Errors and warnings	10.9.2	50 ms	An Event with MODE "Event appears" shall stay at least for the duration of this time.
EventCount	8.2.2.11	1	Constraint for AL_Event.req

10.8 IO Device description (IODD)

An IODD (I/O Device Description) is a file that provides all the necessary properties to establish communication and the necessary parameters and their boundaries to establish the desired function of a sensor or actuator.

An IODD (I/O Device Description) is a file that formally describes a Device.

An IODD file shall be provided for each Device, and shall include all information necessary to support this standard.

The IODD can be used by engineering tools for PLCs and/or Masters for the purpose of identification, configuration, definition of data structures for Process Data exchange, parameterization, and diagnosis decoding of a particular Device.

NOTE Details of the IODD language to describe a Device can be found in [6].

10.9 Device diagnosis

10.9.1 Concepts

This standard provides only most common EventCodes in D.2. It is the purpose of these common diagnosis informations to enable an operator or maintenance person to take fast remedial measures without deep knowledge of the Device's technology. Thus, the text associated with a particular EventCode shall always contain a corrective instruction together with the diagnosis information.

Fieldbus-Master-Gateways tend to only map few EventCodes to the upper system level. Usually, vendor specific EventCodes defined via the IODD can only be decoded into readable instructions via a Port and Device Configuration Tool (PDCT) or specific vendor tool using the IODD.

Condensed information of the Device's "state of health" can be retrieved from the parameter "Device Status" (see B.2.18). Table 98 provides an overview of the various possibilities for Devices and shows examples of consumers for this information.

If implemented, it is also possible to read the number of faults since power-on or reset via the parameter "Error Count" (see B.2.17) and more information in case of profile Devices via the parameter "Detailed Device Status" (see B.2.19).

NOTE Profile specific values for the "Detailed Device Status" are given in [7].

If required, it is highly recommended to provide additional "deep" technology specific diagnosis information in the form of Device specific parameters (see Table B.8) that can be retrieved via port and Device configuration tools for Masters or via vendor specific tools. Usually, only experts or service personnel of the vendor are able to draw conclusions from this information.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 98 – Classification of Device diagnosis incidents

Diagnosis incident	Appear/ disappear	Single shot	Parameter	Destination	Consumer
Error (fast remedy; standard EventCodes)	yes	-	-	PLC or HMI (fieldbus mapping)	Maintenance and repair personnel
Error (IODD: vendor specific EventCodes; see Table D.1)	yes	-	-	PDCT or vendor tool	Vendor service personnel
Error (via Device specific parameters)	-	-	See Table B.8	PDCT or vendor tool	Vendor service personnel
Warning (fast remedy; standard EventCodes)	yes	-	-	PLC or HMI	Maintenance and repair personnel
Warning (IODD: vendor specific EventCodes; see Table D.1)	yes	-		PDCT or vendor tool	Vendor service personnel
Warning (via Device specific parameters)	-	-	See Table B.8		
Notification (Standard EventCodes)	-	yes		PDCT	Commissioning personnel
Detailed Device status	-	-		PDCT or vendor tool	Commissioning personnel and vendor service personnel
Number of faults via parameter "Error Count"	-	-	See B.2.17		
Device "health" via parameter "Device Status"	-	-	See B.2.18, Table B.13	HMI, Tools such as "Asset Management"	Operator

10.9.2 Events

MODE values shall be assigned as follows (see A.6.4):

- Events of TYPE "Error" shall use the MODEs "Event appears / disappears";
- Events of TYPE "Warning" shall use the MODEs "Event appears / disappears";
- Events of TYPE "Notification" shall use the MODE "Event single shot".

The following requirements apply.

- All Events already placed in the Event queue are discarded by the Event Dispatcher when communication is interrupted or cancelled.

  NOTE After communication resumes, the technology specific application is responsible for proper reporting of the current Event causes.

- It is the responsibility of the Event Dispatcher to control the "Event appears" and "Event disappears" flow. Once the Event Dispatcher has sent an Event with MODE "Event appears" for a given EventCode, it shall not send it again for the same EventCode before it has sent an Event with MODE "Event disappears" for this same EventCode.
- Each Event shall use static mode, type, and instance attributes.
- Each vendor specific EventCode shall be uniquely assigned to one of the TYPEs (Error, Warning, or Notification).

In order to prevent the diagnosis communication channel (see Figure 6) from being flooded, the following requirements apply.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- The same diagnosis information shall not be reported at less than 60 s intervals, that is the Event Dispatcher shall not invoke the AL_Event service with the same EventCode more often than 60 s.
- The Event Dispatcher shall not issue an "Event disappears" less than 50 ms after the corresponding "Event appears".
- Subsequent incidents of errors or warnings with the same root cause shall be disregarded, that means one root cause shall lead to a single error or warning.
- The Event Dispatcher shall not invoke the AL_Event service with an EventCount greater than one.
- Errors are prioritized over Warnings.

Figure 91 shows how two successive errors are processed, and the corresponding flow of "Event appears" / "Event disappears" Events for each error.

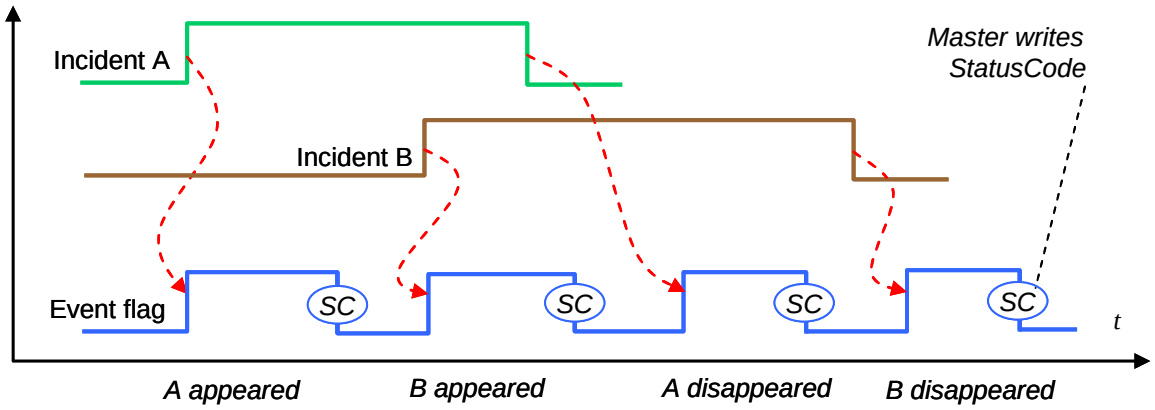


Figure 91 – Event flow in case of successive errors

10.9.3 Visual indicators

The indication of SDCI communication on the Device is optional. The SDCI indication shall use a green indicator. The indication follows the timing and specification shown in Figure 92.

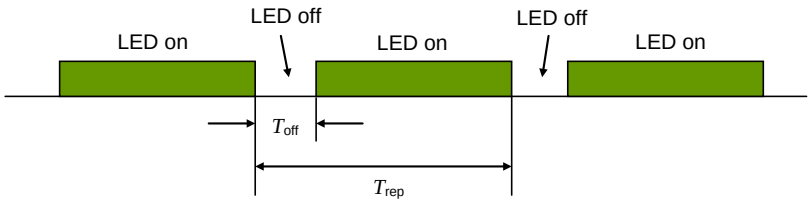


Figure 92 – Device LED indicator timing

Table 99 defines the timing for the LED indicator of Devices.

Table 99 – Timing for LED indicators

Timing	Minimum	Typical	Maximum	Unit
t_{rep}	750	1 000	1 250	ms
T_{off}	75	100	150	ms
T_{off} /T_{rep}	7,5	10	12,5	%

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE Timings above are defined such that the general perception would be "power is on".

A short periodical interruption indicates that the Device is in COMx communication state. In order to avoid flickering, the indication cycle shall start with a "LED off" state and shall always be completed (see Table 99).

10.10 Device connectivity

See 5.5 for the different possibilities of connecting Devices to Master ports and the corresponding cable types as well as the color coding.

NOTE For compatibility reasons, this standard does not prevent SDCI devices from providing additional wires for connection to functions outside the scope of this standard (for example to transfer analog output signals).

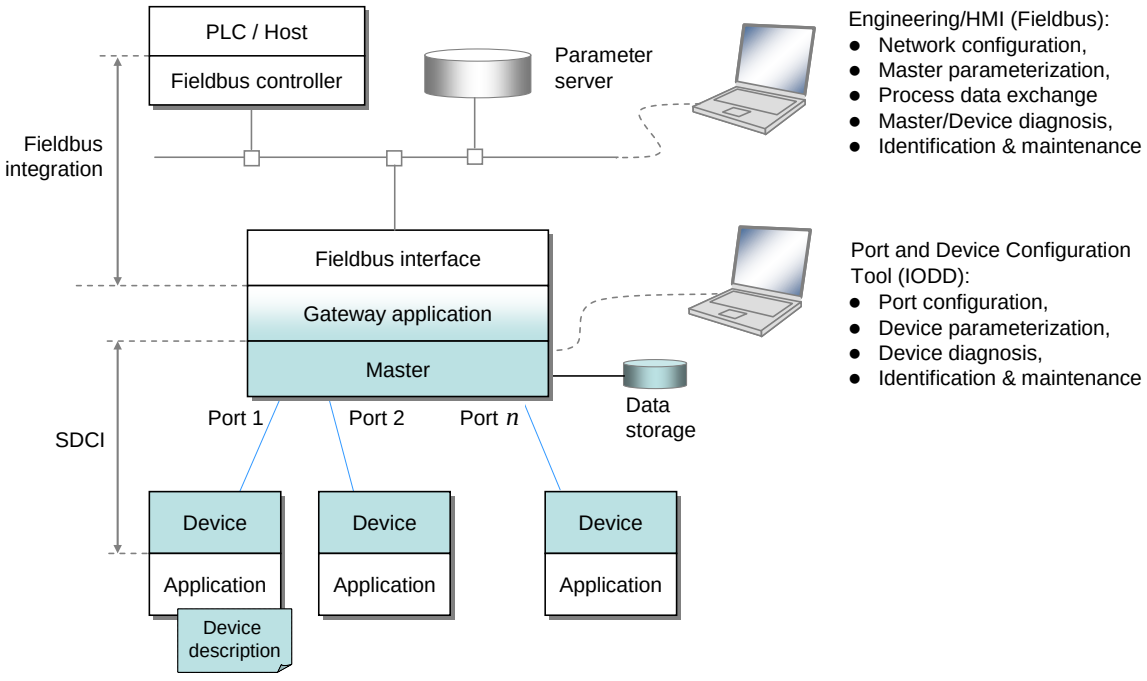
11 Master

11.1 Overview

11.1.1 Generic model for the system integration of a Master

In 4.2 the domain of the SDCI technology within the automation hierarchy is already illustrated.

Figure 93 shows the recommended relationship between the SDCI technology and a fieldbus technology. Even though this may be the major use case in practice, this does not automatically imply that the SDCI technology depends on the integration into fieldbus systems. It can also be directly integrated into PLC systems, industrial PC, or other control systems without fieldbus communication in between.



NOTE Blue shaded areas indicate features specified in this standard.

Figure 93 – Generic relationship of SDCI technology and fieldbus technology

11.1.2 Structure and services of a Master

Figure 94 provides an overview of the complete structure and the services of a Master.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The Master applications comprise first a fieldbus specific gateway or direct connection to a PLC (host) for the purpose of start-up configuration and parameterization as well as Process Data exchange, user-program-controlled parameter change at runtime, and diagnosis propagation. For the purpose of configuration, parameterization, and diagnosis during commissioning a so-called "Port and Device Configuration Tool" (Software) is connected either directly to the Master or via fieldbus communication. These two instruments are using the following common Master applications:

- Configuration Manager (CM), which transforms the user configuration assignments into port set-ups;
- On-request Data Exchange (ODE), which provides for example acyclic parameter access;
- Data Storage (DS) mechanism, which can be used to save and restore the Device parameters;
- Diagnosis Unit (DU), which routes Events from the AL to the Data Storage unit or the gateway application;
- Process Data Exchange (PDE), building the bridge to upper level automation instruments.

These Master applications provide standard methods/functions common to all Masters.

The Configuration Manager (CM) and the Data Storage mechanism (DS) need special coordination in respect to On-request Data, see Figure 95 and Figure 105.

The gateway application maps these functions into the features of a particular fieldbus/PLC or directly into a host system. It is not within the scope of this standard to define any of these gateway applications.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

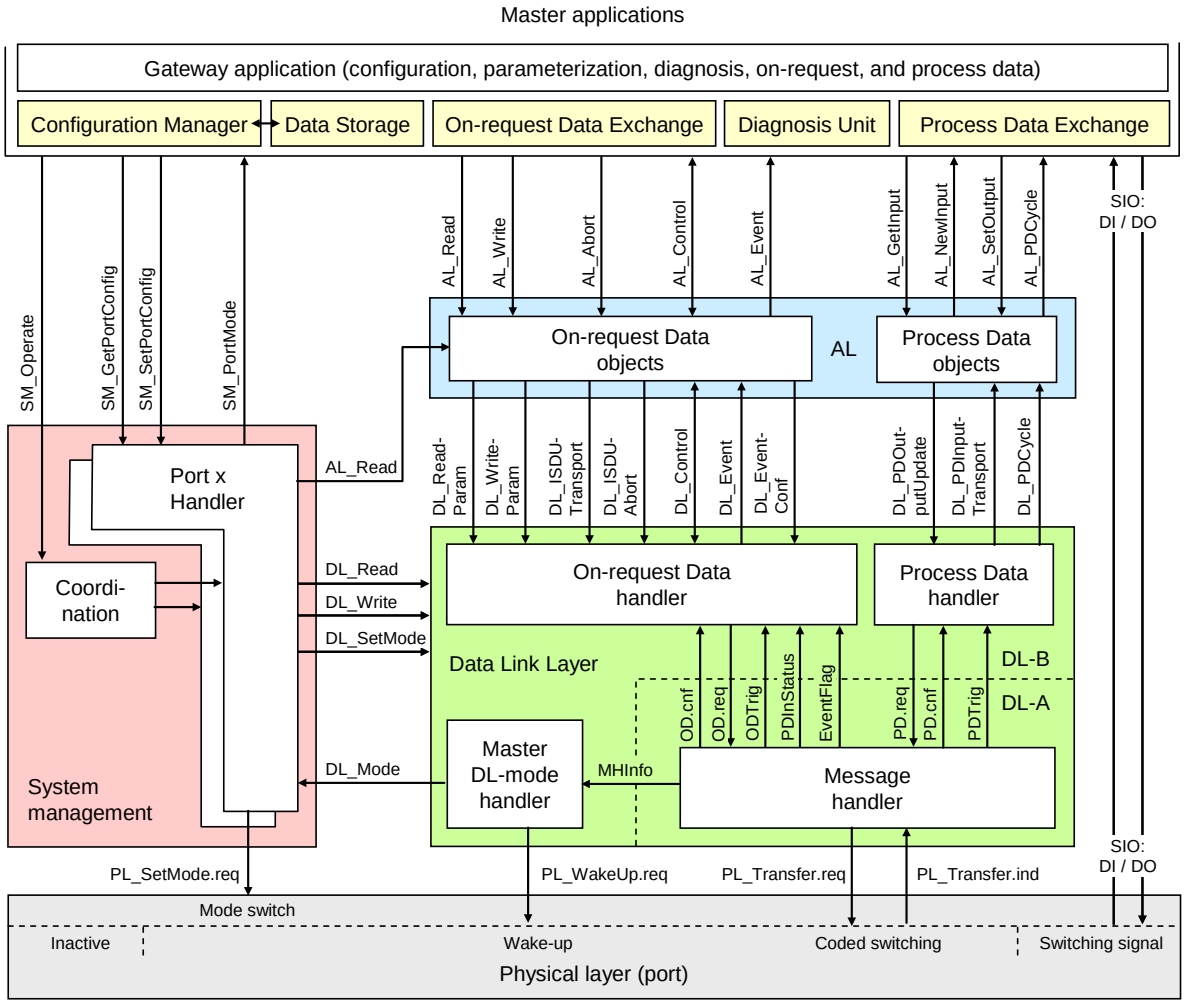


Figure 94 – Structure and services of a Master

Figure 95 shows the relationship of the common Master applications.

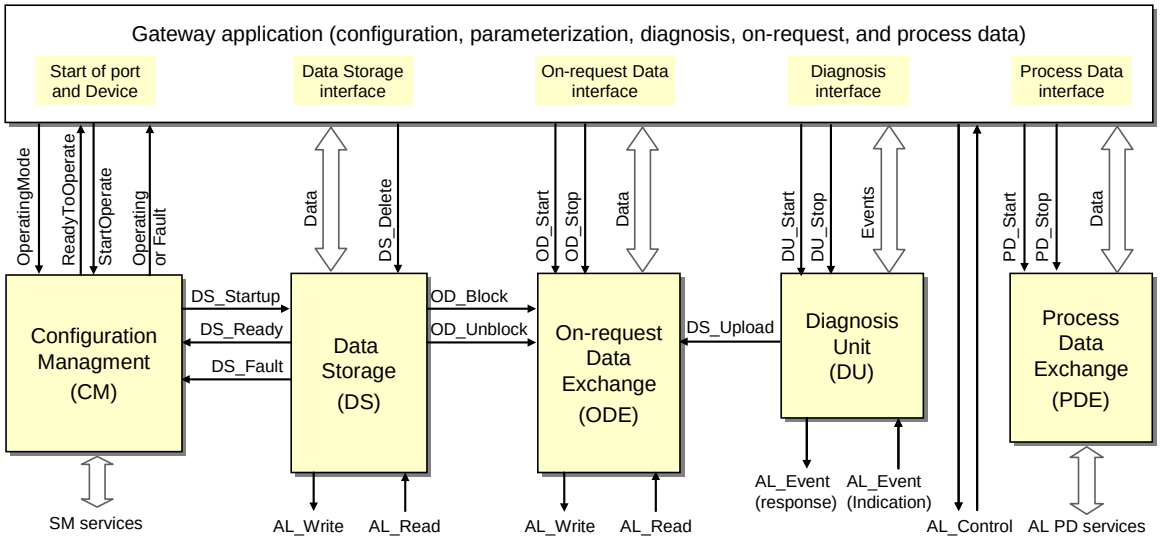


Figure 95 – Relationship of the common Master applications

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The internal variables between the common Master applications are specified in Table 100. The main responsibility is assigned to the Configuration Manager (CM) as shown in Figure 95 and explained in 11.2.

Table 100 – Internal variables and Events to control the common Master applications

Internal Variable	Definition
OperatingMode	This variable activates the port and provides the configuration parameters.
ReadyToOperate	This variable indicates correct configuration of the port.
StartOperate	This variable allows for explicite change of all ports to the OPERATE mode.
Operating	This variable indicates all ports are in cyclic Process Data exchange mode
Fault	This variable indicates abandoned COMx communication at any port (see Figure 98 and Table 101).
DS_Startup	This variable triggers the Data Storage (DS) state machine causing an Upload or Download of Device parameters if required (see 11.3).
DS_Ready	This variable indicates the Data Storage has been accomplished successfully; operating mode is CFGCOM or AUTOCOM (see 9.2.2.2)
DS_Fault	This variable indicates the Data Storage has been aborted due to a fault.
DS_Delete	Any verified change of Device configuration leads to a deletion of the stored data set in the Data Storage.
DS_Upload	This variable triggers the Data Storage state machine in the Master due to the special Event "DS_UPLOAD_REQ" from the Device.
OD_Start	This variable enables On-request Data access via AL_Read and AL_Write.
OD_Stop	This variable indicates that On-request Data access via AL_Read and AL_Write is acknowledged with a negative response to the gateway application.
OD_Block	Data Storage upload and download actions disable the On-request Data access through AL_Read or AL_Write. Access by the gateway application is denied.
OD_Unblock	This variable enables On-request Data access via AL_Read or AL_Write.
DU_Start	This variable enables the Diagnosis Unit to propagate remote (Device) or local (Master) Events to the gateway application.
DU_Stop	This variable indicates that the Device Events are not propagated to the gateway application and not acknowledged. Available Events are blocked until the DU is enabled again.
PD_Start	This variable enables the Process Data exchange with the gateway application.
PD_Stop	This variable disables the Process Data exchange with the gateway application.

11.2 Configuration Manager (CM)

11.2.1 General

Figure 95 and Figure 96 demonstrate the coordinating role of the configuration manager amongst all the common Master applications. After setting up a port to the assigned modes (see 11.2.2.1 through 11.2.2.3) CM starts the Data Storage mechanism (DS) and returns the variable "Operating" or "Fault" to the gateway application.

In case of the variable "Operating" of a particular port, the gateway application activates the state machines of the associated Diagnosis Unit (DU), the On-request Data Exchange (ODE), and the Process Data Exchange (PDE).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

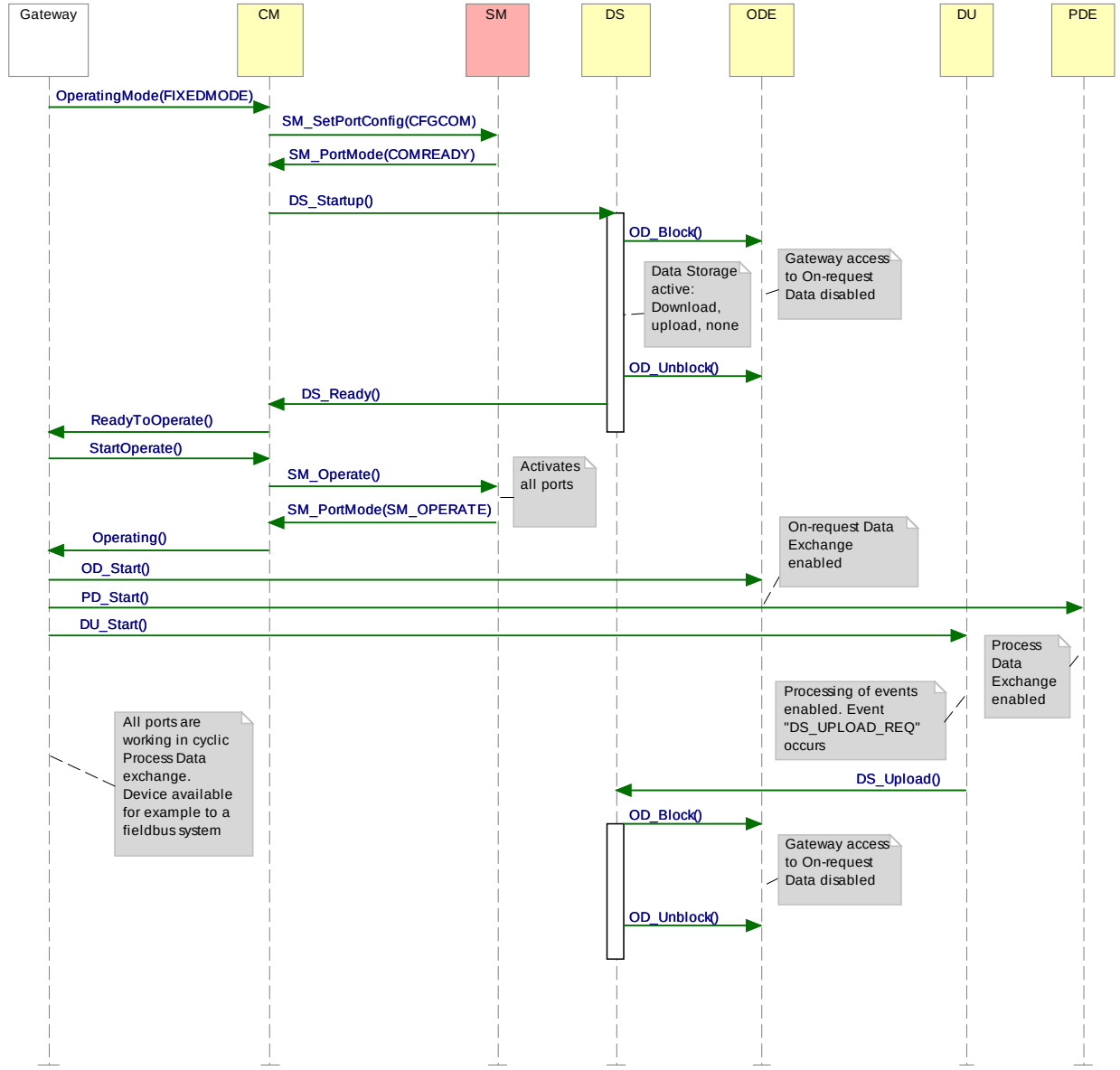


Figure 96 – Sequence diagram of configuration manager actions

After all SDCI ports are ready ("ReadyToOperate", see Figure 96), the gateway application shall activate all ports ("StartOperate") to ensure that synchronization of port cycles can take place. Finally, the Devices are exchanging Process Data ("Operating"). In case of faults the gateway application receives "Communication abandoned" ("INACTIVE" or "COMLOST").

In case of SM_PortMode (COMP_FAULT, REVISION_FAULT, or SERNUM_FAULT) according to 9.2.3, only the ODE machine shall be activated to allow for parameterization.

At each new start of a port the gateway application will first de-activate (e.g. OD_Stop) the associated machines DU, ODE, and PDE.

Several parameters are available for the configuration manager to achieve a specific behaviour.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.2.2 Configuration parameter

11.2.2.1 OperatingMode

One of the following operating modes can be selected. All modes are mandatory.

INACTIVE
The SDCI port is deactivated, the corresponding Process Data length for input and output is zero. The Master shall not have any activities on this port.

DO
The SDCI port is configurated as a digital output (see Table 2 for constraints). The output Process Data length is 1 bit. The Master shall not try to wake up any Device at this port.

DI
The SDCI port is configurated as a digital input. The input Process Data length is 1 bit. The Master shall not try to wake up any Device at this port.

FIXEDMODE
An SDCI port is configured for continuous communication. The defined identification is checked. Whether a difference in Device identification will lead to the rejection of the Device or not depends on the port configuration (InspectionLevel, see Table 78).

SCANMODE
The SDCI port is configured for continuous communication. The identification is read back from the Device and can be provided as the new defined identification. Otherwise see OperatingMode "FIXEDMODE".

11.2.2.2 PortCycle

One of the following port cycle modes can be selected. None of the modes is mandatory.

FreeRunning
The port cycle timing is not restricted.

FixedValue
The port cycle timing is fixed to a specific value. If the Device is not able to achieve this timing, for example if the timing is lower than the MinCycleTime of the Device, an error shall be generated. The fixed value can be written in the CycleTime parameter as specified in 11.2.2.3.

MessageSync
The port cycle timing is restricted to the synchronous start of all messages on all SDCI ports of this Master. In this case the cycle time is given by the highest MinCycleTime of the connected Devices. All Master ports set to this mode are working with this behaviour as shown in Figure 97. Values for displacement and jitter shall be noted in the user manual.

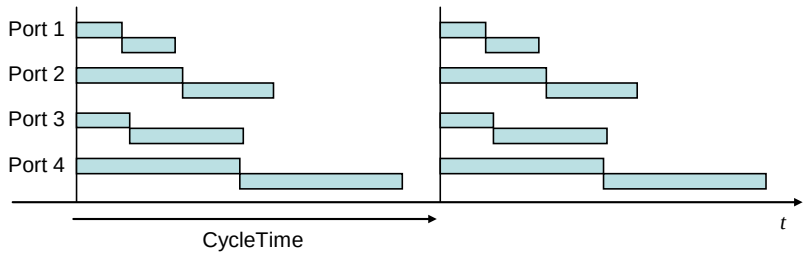


Figure 97 – Ports in MessageSync mode

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.2.2.3 CycleTime

This parameter contains the requested or actual cycle time for the specific ports. It shall be passed as a value with a resolution of 100 μs.

11.2.2.4 PDConfig

This set of parameters contains the rules for the Process Data mapping between the Device Process Data stream and the gateway Process Data stream (see example in Figure 107 for the definitions).

LenIn
This parameter contains the requested length of the Device input ProcessData in Bits.

PosIn
This parameter contains the offset within the gateway input Process Data stream in Bit.

SrcOffsetIn
This parameter contains the offset within the Device input Process Data stream in Bit.

LenOut
This parameter contains the requested length of the Device output ProcessDataOut Bits.

PosOut
This parameter contains the offset within the gateway output Process Data stream in Bit.

SrcOffsetOut
This parameter contains the offset within the Device output Process Data stream in Bit.

11.2.2.5 DeviceIdentification

This set of parameters contains the actual configured Device identification.

VendorID
This parameter contains the requested or read vendor specific ID as specified in B.1.8.

DeviceID
This parameter contains the requested or read Device specific ID as specified in B.1.9.

SerialNumber

This parameter contains the requested or read SerialNumber as specified in B.2.13.

InspectionLevel

This parameter contains the requested InspectionLevel as specified in Table 78.

11.2.2.6 DataStorageConfig

This set of parameter items contains the settings of the Data Storage (DS) mechanism.

ActivationState
This parameter contains the requested state of the DS mechanism for this port. The following modes are supported:

DS_Enabled
The DS mechanism is active and provides the full functionality as specified in 11.3.2.

DS_Disabled
The DS mechanism is inactive and the complete parameter set of this port remains stored.

DS_Cleared
The DS mechanism is disabled and the stored parameter set of this port is cleared.

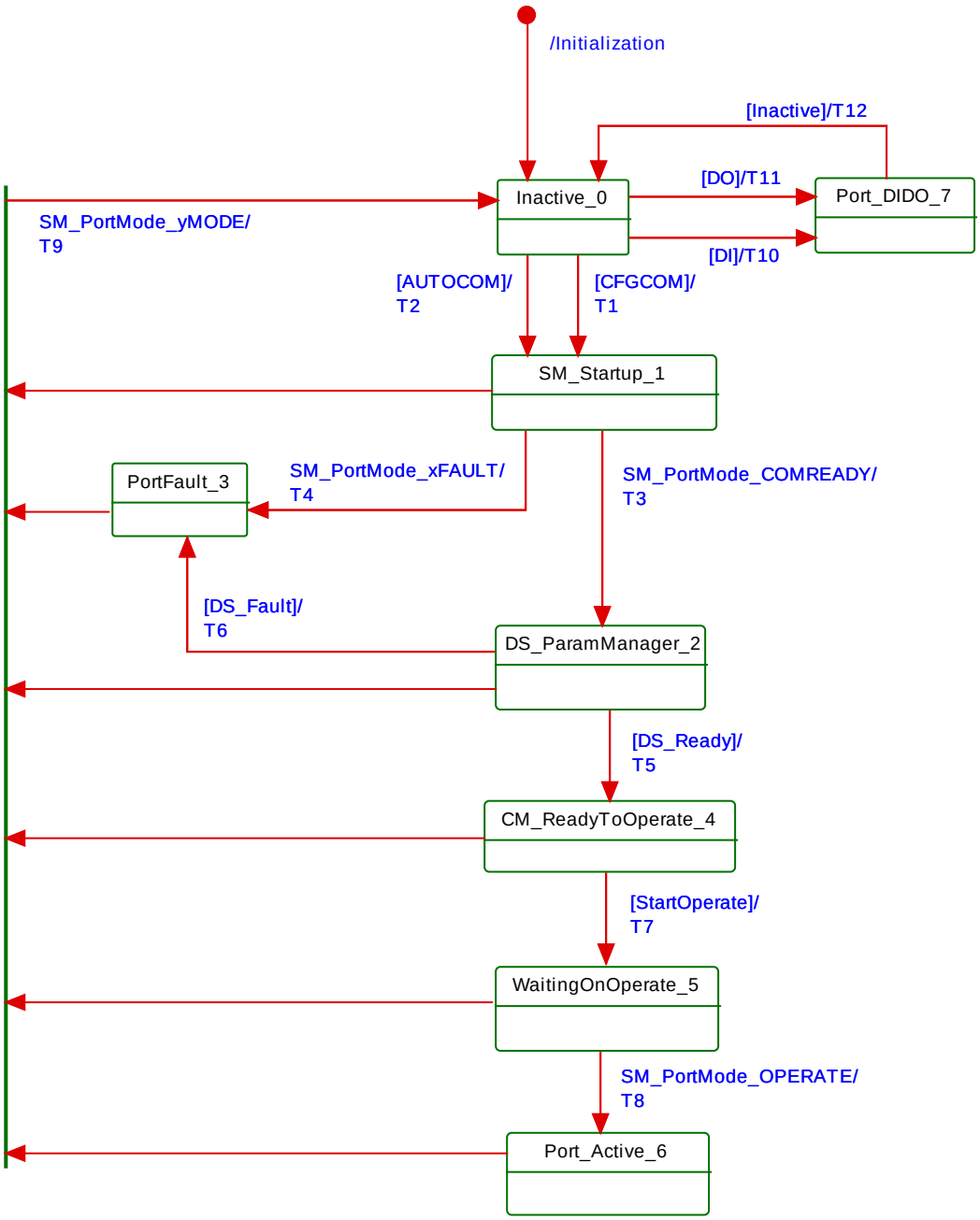
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DownloadEnable
The DS mechanism is permitted to write data to the connected Device.

UploadEnable
The DS mechanism is permitted to read data from the connected Device.

11.2.3 State machine of the Configuration Manager

Figure 98 shows the state machine of the Master configuration manager.



Key
xFAULT: REV_FAULT or COMP_FAULT or SERNUM_FAULT
yMODE: INACTIVE or COMLOST

Figure 98 – State machine of the Configuration Manager

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The different states show the steps of necessary commands to establish or maintain communication or the DI or DO state.

Any change of the port configuration can be activated by changing the OperatingMode variable (see 11.2.2.1).

Table 101 shows the state transition table of the configuration manager state machine.

Table 101 – State transition tables of the Configuration Manager

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting on any of the OperatingMode variables from the gateway application: DO, DI, AUTOCOM, or CFGCOM.
SM_Startup_1	Waiting on an established communication or loss of communication or any of the faults REVISION_FAULT, COMP_FAULT, or SERNUM_FAULT (see Table 83).
DS_ParamManager_2	Waiting on accomplished Data Storage startup. Parameter are downloaded into the Device or uploaded from the Device.
PortFault_3	Device in state PREOPERATE (communicating). However, one of the three faults REVISION_FAULT, COMP_FAULT, SERNUM_FAULT, or DS_Fault occurred.
CM_ReadytoOperate_4	Port is waiting until the gateway application indicates "StartOperate".
WaitingOnOperate_5	Waiting on SM to switch to OPERATE.
PortActive_6	Port is in OPERATE mode. The gateway application is exchanging Process Data and ready to send or receive On-request Data.
PortDIDO_7	Port is in DI or DO mode. The gateway application is exchanging Process Data (DI or DO).

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	SM_SetPortConfig_CFGCOM
T2	0	1	SM_SetPortConfig_AUTOCOM
T3	1	2	DS_Startup: The DS state machine is triggered.
T4	1	3	"Fault" indication to gateway application (REVISION_FAULT, COMP_FAULT, or SERNUM_FAULT), see Figure 95.
T5	2	4	Indication to gateway application: ReadyToOperate
T6	2	3	Data Storage failed. Rollback to previous parameter set.
T7	4	5	SM_Operate.
T8	5	6	Indication to gateway application: "Operating" (see Figure 96).
T9	1,2,3,4,5, 6	0	SM_SetPortConfig_INACTIVE. "Fault" indication to gateway application: COMLOST or INACTIVE
T10	0	7	SM_SetPortConfig_DI. Indication to gateway application: DI
T11	0	7	SM_SetPortConfig_DO. Indication to gateway application: DO
T12	7	0	SM_SetPortConfig_INACTIVE.

INTERNAL ITEMS	TYPE	DEFINITION
DS_Ready	Bool	Data Storage sequence (upload, download) accomplished. Port operating mode is FIXEDMODE or SCANMODE. See Table 100.
DS_Fault	Bool	See Table 100.
StartOperate	Bool	Gateway application causes the port to switch to OPERATE.
FIXEDMODE	Bool	One of the OperatingModes (see 11.2.2.1)
SCANMODE	Bool	One of the OperatingModes (see 11.2.2.1)
DI	Bool	One of the OperatingModes (see 11.2.2.1)
DO	Bool	One of the OperatingModes (see 11.2.2.1)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.3 Data Storage (DS)

11.3.1 Overview

Data Storage between Master and Device is specified within this standard, whereas the adjacent upper Data Storage mechanisms depend on the individual fieldbus or system. The Device holds a standardized set of objects providing parameters for Data Storage, memory size requirements, control and state information of the Data Storage mechanism. Changes of Data Storage parameter sets are detectable via the "Parameter Checksum" (see 10.4.8).

11.3.2 DS data object

The structure of a Data Storage data object is specified in Table F.1.

The Master shall always hold the header information (Parameter Checksum, VendorID, and DeviceID) for the purpose of checking and control. The object information (objects 1...n) will be stored within the non-volatile memory part of the Master (see Annex F). Prior to a download of the Data Storage data object (parameter block), the Master will check the consistency of the header information with the particular Device.

The maximum permitted size of the Data Storage data object is 2×2^{10} octets. It is mandatory for Masters to provide at least this memory space per port if the Data Storage mechanism is implemented.

11.3.3 DS state machine

The Data Storage mechanism is called right after establishing the COMx communication, before entering the OPERATE mode. During this time any other communication with the Device shall be rejected by the gateway.

Figure 99 shows the state machine of the Data Storage mechanism.

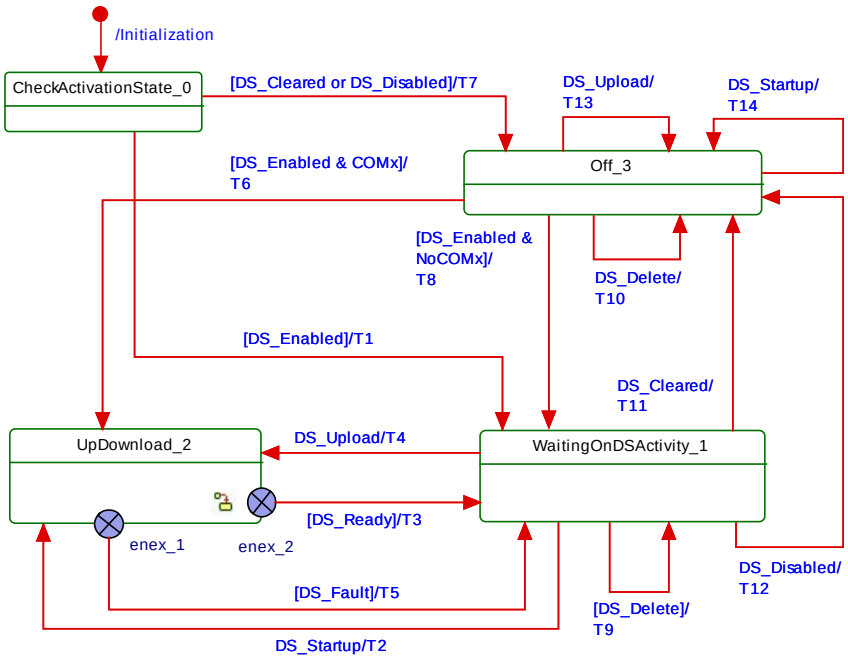


Figure 99 – Main state machine of the Data Storage mechanism

Figure 100 shows the submachine of the state "UpDownload_2".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

This submachine can be invoked by the Data Storage mechanism or during runtime triggered by a "DS_UPLOAD_REQ" Event.

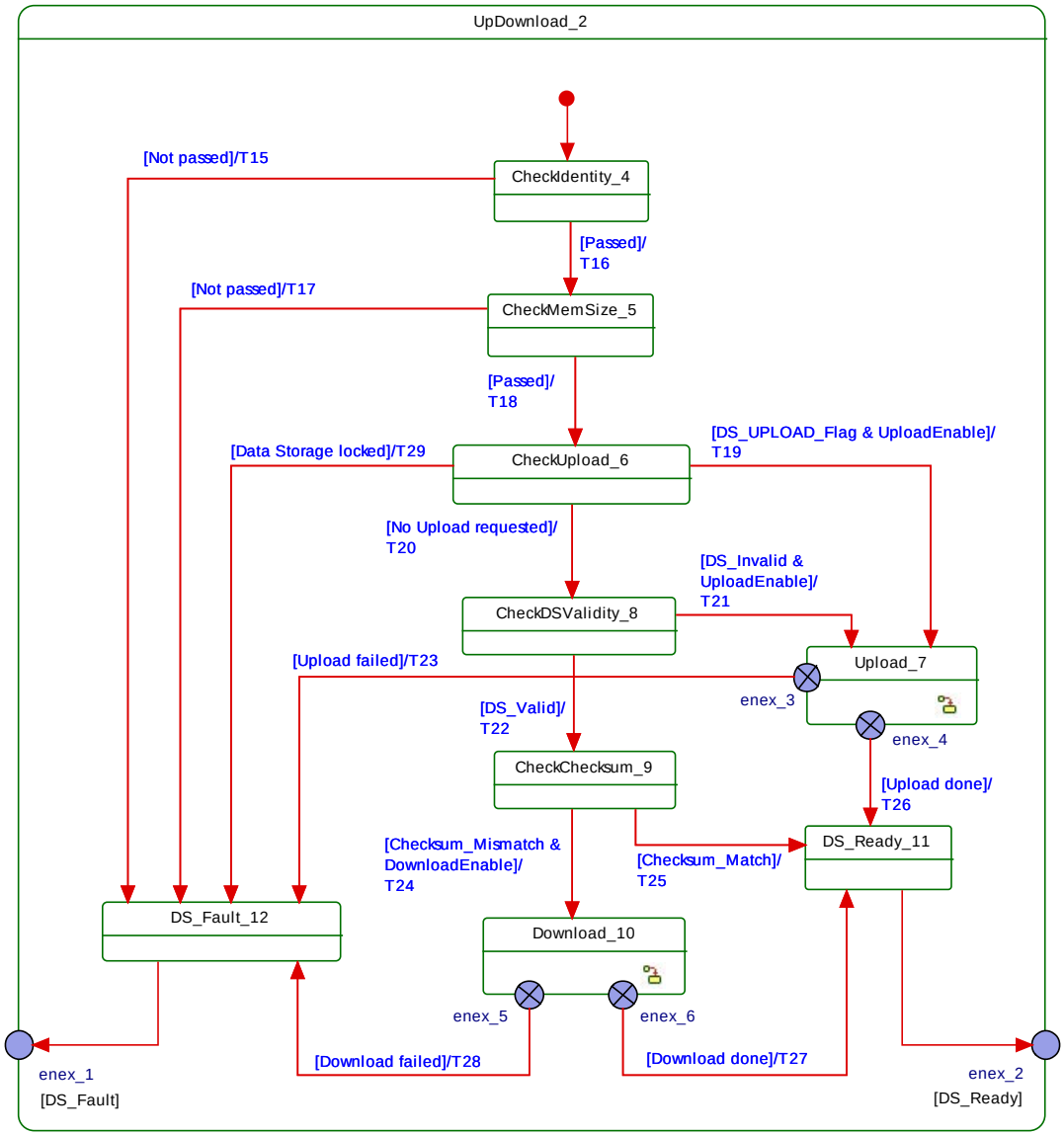


Figure 100 – Submachine "UpDownload_2" of the Data Storage mechanism

Figure 101 shows the submachine of the state "Upload_7".

This state machine can be invoked by the Data Storage mechanism or during runtime triggered by a DS_UPLOAD_REQ Event.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

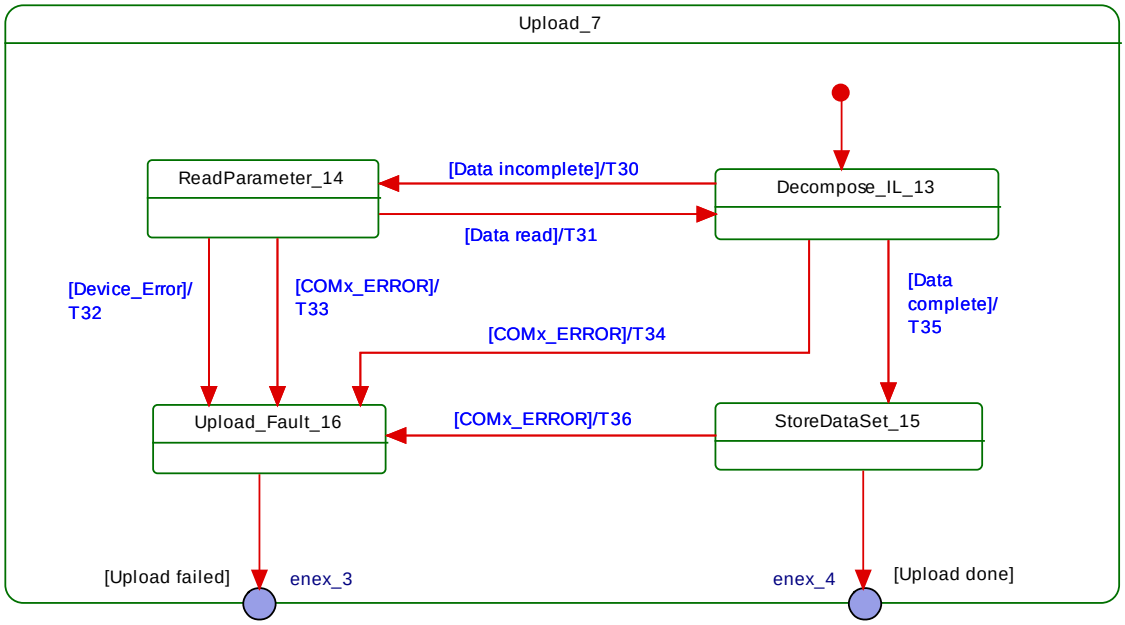


Figure 101 – Data Storage submachine "Upload_7"

Figure 102 demonstrates the Data Storage upload sequence using the Data Storage Index (DSI) specified in B.2.3 and Table B.10. The structure of Index_List is specified in Table B.11. The DS_UPLOAD_FLAG shall be reset at the end of each sequence (see Table B.10).

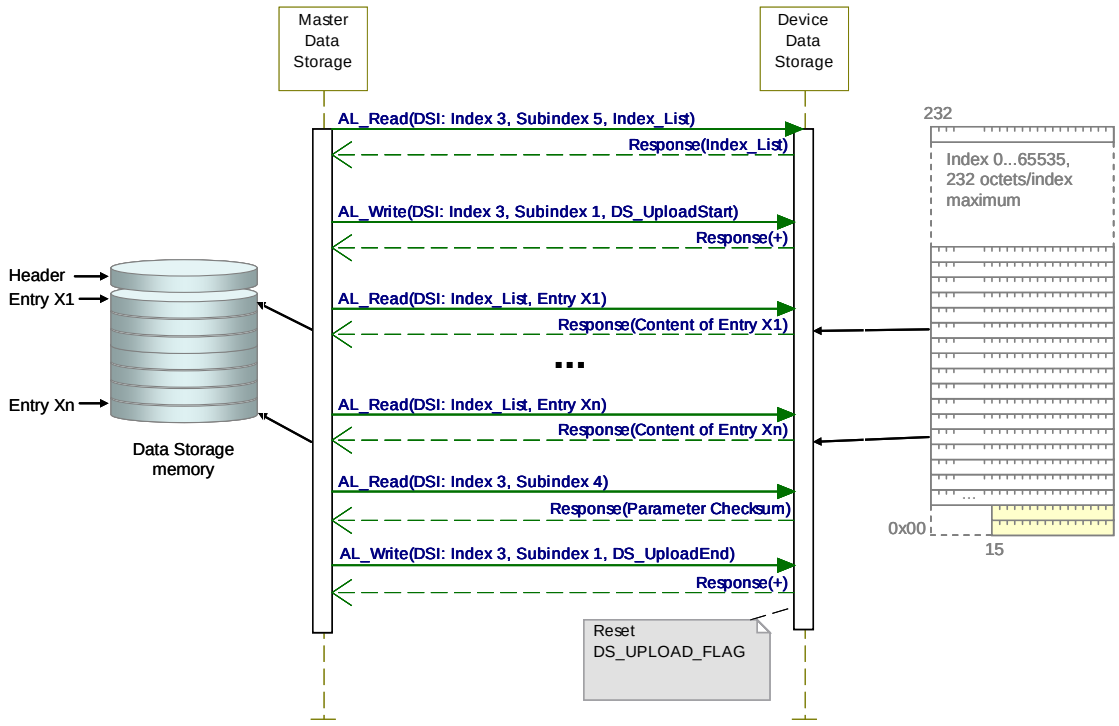


Figure 102 – Data Storage upload sequence diagram

Figure 103 shows the submachine of the state "Download_10".

This state machine can be invoked by the Data Storage mechanism.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

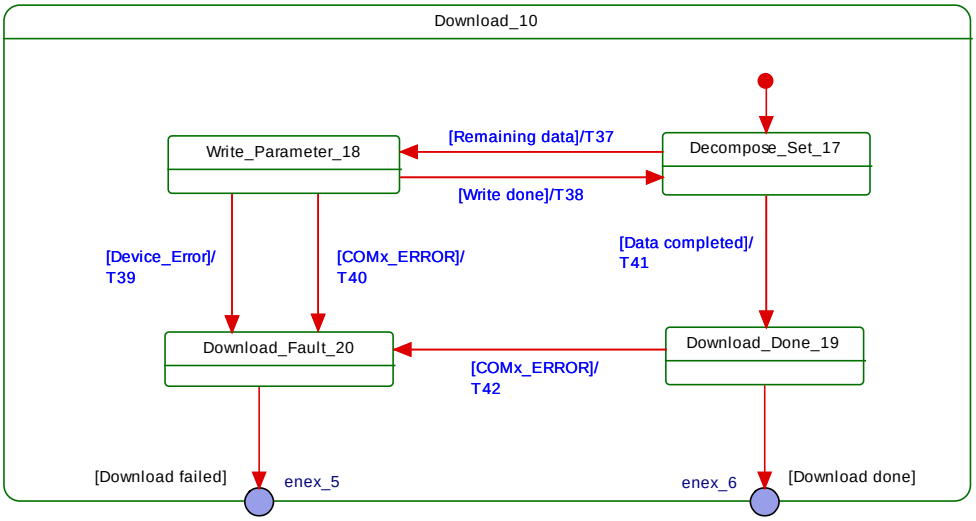


Figure 103 – Data Storage submachine "Download_10"

Figure 104 demonstrates the Data Storage download sequence using the Data Storage Index (DSI) specified in B.2.3 and Table B.10. The structure of Index_List is specified in Table B.11. The DS_UPLOAD_FLAG shall be reset at the end of each sequence (see Table B.10).

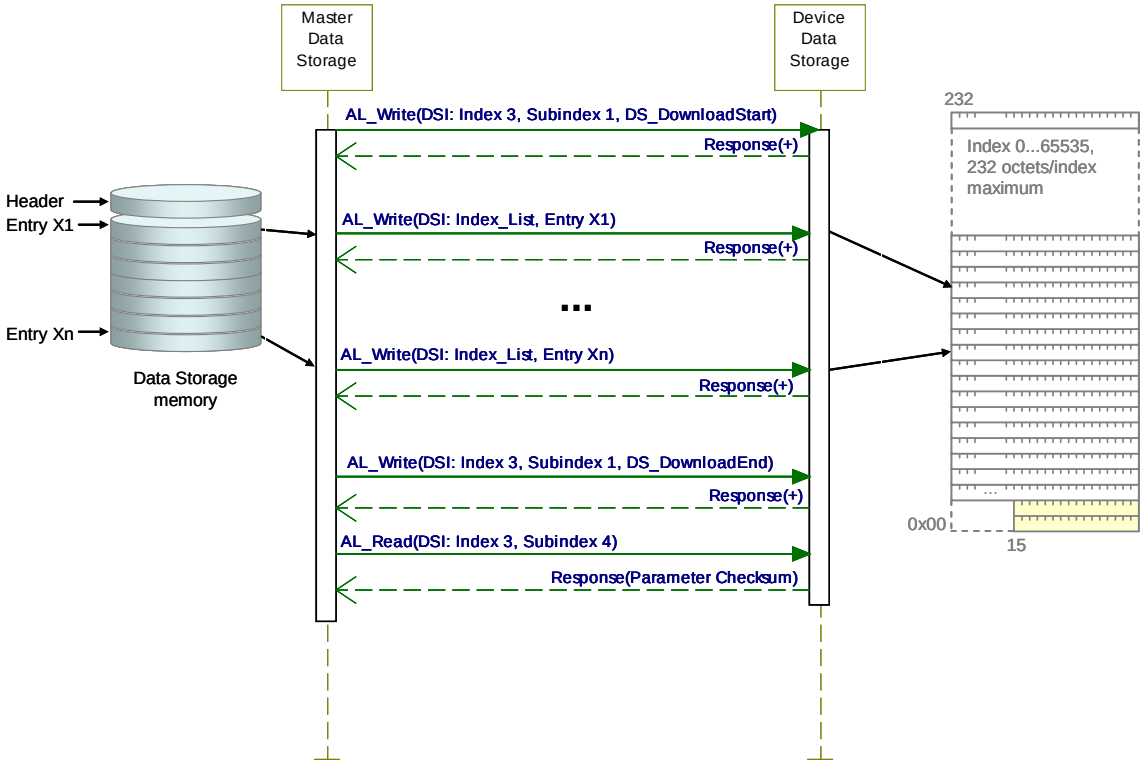


Figure 104 – Data Storage download sequence diagram

Table 102 shows the states and transitions of the Data Storage state machines.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table 102 – States and transitions of the Data Storage state machines

STATE NAME	STATE DESCRIPTION
CheckActivationState_0	Check current state of the DS configuration: Independently from communication status, DS_Startup from configuration management or an Event DS_UPLOAD_REQ is expected.
WaitingOnDSActivity_1	Waiting for upload request, Device startup, all changes of activation state independent of the Device communication state.
UpDownload_2	Submachine for up/download actions and checks
Off_3	Data Storage handling switched off or deactivated
SM: CheckIdentity_4	Check Device identification (DeviceID, VendorID) against parameter set within the Data Storage (see Table F.2). Empty content does not lead to a fault.
SM: CheckMemSize_5	Check data set size (Index 3, Subindex 3) against available Master storage size
SM: CheckUpload_6	Check for DS_UPLOAD_FLAG within the Data Storage Index (see Table B.10)
SM: Upload_7	Submachine for the upload actions
SM: CheckDSValidity_8	Check whether stored data within the Master is valid or invalid. A Master could be replaced between upload and download activities. It is the responsibility of a Master designer to implement a validity mechanism according to the chosen use cases
SM: CheckChecksum_9	Check for differences between the data set content and the Device parameter via the "Parameter Checksum" within the Data Storage Index (see Table B.10)
SM: Download_10	Submachine for the download actions
SM: DS_Ready_11	Prepare DS_Ready indication to the Configuration Management (CM)
SM: DS_Fault_12	Prepare DS_Fault indication from "Identification_Fault", "SizeCheck_Fault", "Upload_Fault", and "Download_Fault" to the Configuration Management (CM)
SM: Decompose_IL_13	Read Index List within the Data Storage Index (see Table B.10). Read content entry by entry of the Index List from the Device (see Table B.11).
SM: ReadParameter_14	Wait until read content of one entry of the Index List from the Device is accomplished.
SM: StoreDataSet_15	Task of the gateway application: store entire data set according to Table F.1 and Table F.2
SM: Upload_Fault_16	Prepare Upload_Fault indication from "Device_Error" and "COM_ERROR" as input for the higher level indication DS_Fault.
SM: Decompose_Set_17	Write parameter by parameter of the data set into the Device according to Table F.1.
SM: Write_Parameter_18	Wait until write of one parameter of the data set into the Device is accomplished.
SM: Download_Done_19	Download completed. Read back "Parameter Checksum" from the Data Storage Index according to Table B.10. Save this value in the stored data set according to Table F.2.
SM: Download_Fault_20	Prepare Download_Fault indication from "Device_Error" and "COM_ERROR" as input for the higher level indication DS_Fault.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	2	-
T3	2	1	OD_Unblock; Indicate DS_Ready to CM
T4	1	2	Confirm Event "DS_UPLOAD_REQ"
T5	2	1	DS_Break (AL_Write, Index 3, Subindex 1); clear intermediate data (garbage collection); rollback to previous parameter state; DS_Fault (see Figure 95); OD_Unblock.
T6	3	2	-
T7	0	3	-
T8	3	1	-
T9	1	1	Clear saved parameter set (see Table F.1 and Table F.2)
T10	3	3	Clear saved parameter set (see Table F.1 and Table F.2)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T11	1	3	Clear saved parameter set (see Table F.1 and Table F.2)
T12	1	3	-
T13	3	3	Confirm Event "DS_UPLOAD_REQ"; no further action
T14	3	3	DS_Ready to CM
T15	4	12	Indicate DS_Fault(Identification_Fault) to the gateway application
T16	4	5	Read "Data Storage Size" according to Table B.10, OD_Block
T17	5	12	Indicate DS_Fault(SizeCheck_Fault) to the gateway application
T18	5	6	Read "DS_UPLOAD_FLAG" according to Table B.10
T19	6	7	Data Storage Index 3, Subindex 1: "DS_UploadStart" (see Table B.10)
T20	6	8	-
T21	8	7	Data Storage Index 3, Subindex 1: "DS_UploadStart" (see Table B.10)
T22	8	9	-
T23	7	12	Data Storage Index 3, Subindex 1: "DS_Break" (see Table B.10). Indicate "DS_Fault(Upload)" to the gateway application
T24	9	10	Data Storage Index 3, Subindex 1: "DS_DownloadStart" (see Table B.10)
T25	9	11	-
T26	7	11	Data Storage Index 3, Subindex 1: "DS_UploadEnd"; read Parameter Checksum (see Table B.10)
T27	10	11	-
T28	10	12	Data Storage Index 3, Subindex 1: "DS_Break" (see Table B.10). Indicate "DS_Fault(Download)" to the gateway application.
T29	6	12	Indicate DS_Fault(Data Storage locked) to the gateway application
T30	13	14	AL_Read (Index List)
T31	14	13	-
T32	14	16	-
T33	14	16	-
T34	13	16	-
T35	13	15	Read "Parameter Checksum" (see Table B.10).
T36	15	16	-
T37	17	18	Write parameter via AL_Write
T38	18	17	-
T39	18	20	-
T40	18	20	-
T41	17	19	Data Storage Index 3, Subindex 1: "DS_DownloadEnd" (see Table B.10) Read "Parameter Checksum" (see Table B.10).
T42	19	20	-

INTERNAL ITEMS	TYPE	DEFINITION
DS_Cleared	Bool	Data Storage handling switched off, see 11.2.2.6
DS_Disabled	Bool	Data Storage handling deactivated, see 11.2.2.6
DS_Enabled	Bool	Data Storage handling activated, see 11.2.2.6
COMx_ERROR	Bool	Error in COMx communication detected
Device_Error	Bool	Access to Index denied, AL_Read or AL_Write.cnf(-) with ErrorCode 0x80
DS_Startup	Variable	Trigger from CM state machine, see Figure 95
NoCOMx	Bool	No COMx communication

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTERNAL ITEMS	TYPE	DEFINITION
COMx	Bool	COMx communication working properly
DS_UPLOAD_REQ	Event	See Table D.2
UploadEnable	Bool	Data Storage handling configuration, see 11.2.2.6
DownloadEnable	Bool	Data Storage handling configuration, see 11.2.2.6
DS_Valid	Bool	Valid parameter set available within the Master. See state description "SM: CheckDSValidity_8"
DS_Invalid	Bool	No valid parameter set available within the Master. See state description "SM: CheckDSValidity_8"
Checksum_Mismatch	Bool	Acquired "Parameter Checksum" from Device does not match the checksum within Data Storage (binary comparison)
Checksum_Match	Bool	Acquired "Parameter Checksum" from Devive matches the checksum within Data Storage (binary comparison)

11.3.4 Parameter selection for Data Storage

The Device designer defines the parameters that are part of the Data Storage mechanism.

The IODD marks all parameters not included in Data Storage with the attribute "excludedFromDataStorage". However, the Data Storage mechanism shall not consider the information from the IODD but rather the Parameter List read out from the Device.

11.4 On-Request Data exchange (ODE)

Figure 105 shows the state machine of the Master's On-request Data Exchange. This behaviour is mandatory for a Master.

During an active data transmission of the Data Storage mechanism, all On-request Data requests are blocked.

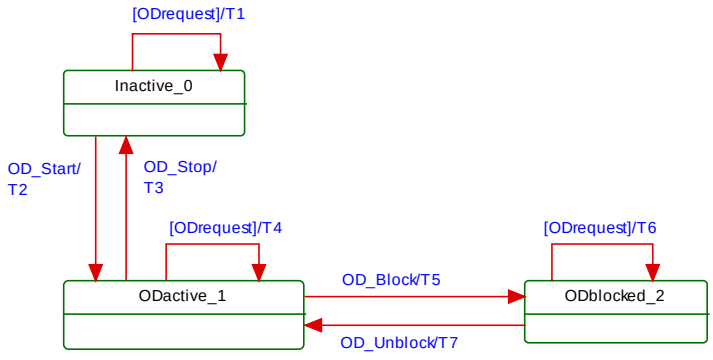


Figure 105 – State machine of the On-request Data Exchange

Table 103 shows the state transition table of the On-request Data Exchange state machine.

Table 103 – State transition table of the ODE state machine

STATE NAME	STATE DESCRIPTION
Inactive_0	Waiting for activation
ODactive_1	On-request Data communication active using AL_Read or AL_Write
ODblocked_2	On-request Data communication blocked

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	0	Access blocked (inactive): indicates "Service not available" to the gateway application
T2	0	1	-
T3	1	0	-
T4	1	1	AL_Read or AL_Write
T5	1	2	-
T6	2	2	Access blocked temporarily: indicates "Service not available" to the gateway application
T7	2	1	-

INTERNAL ITEMS	TYPE	DEFINITION
ODrequest	Variable	On-request Data read or write requested via AL_Read or AL_Write

11.5 Diagnosis Unit (DU)

The Diagnosis Unit (DU) routes Events from the AL to the Data Storage unit or the gateway application. These Events primarily contain diagnosis information.

Main goal for diagnosis information is to alert an operator in an efficient manner. That means:

- no diagnosis information flooding;
- report of the root cause of an incident within a Device or within the Master and no subsequent correlated faults;
- diagnosis information shall provide information on how to maintain or repair the affected component for fast recovery of the automation system.

Within SDCI, diagnosis information of Devices is conveyed to the Master via Events consisting of EventQualifiers and EventCodes (see A.6). The associated human readable text is available for standardized EventCodes within this standard (see Annex D) and for vendor specific EventCodes within the associated IODD file of a Device. The standardized EventCodes can be mapped to semantically identical or closest fieldbus channel diagnosis definitions within the gateway application. Vendor specific IODD codings can be mapped to specific channel diagnosis definitions (individual code and associated human readable information) within the fieldbus device description file.

Fieldbus engineering tools and process monitoring systems (human machine interfaces) can use the fieldbus device description to decode the received fieldbus diagnosis code into human readable diagnosis text.

Diagnosis information flooding is avoided by flow control, which allows for only one Event per Device to be propagated to the Master/gateway application at a time.

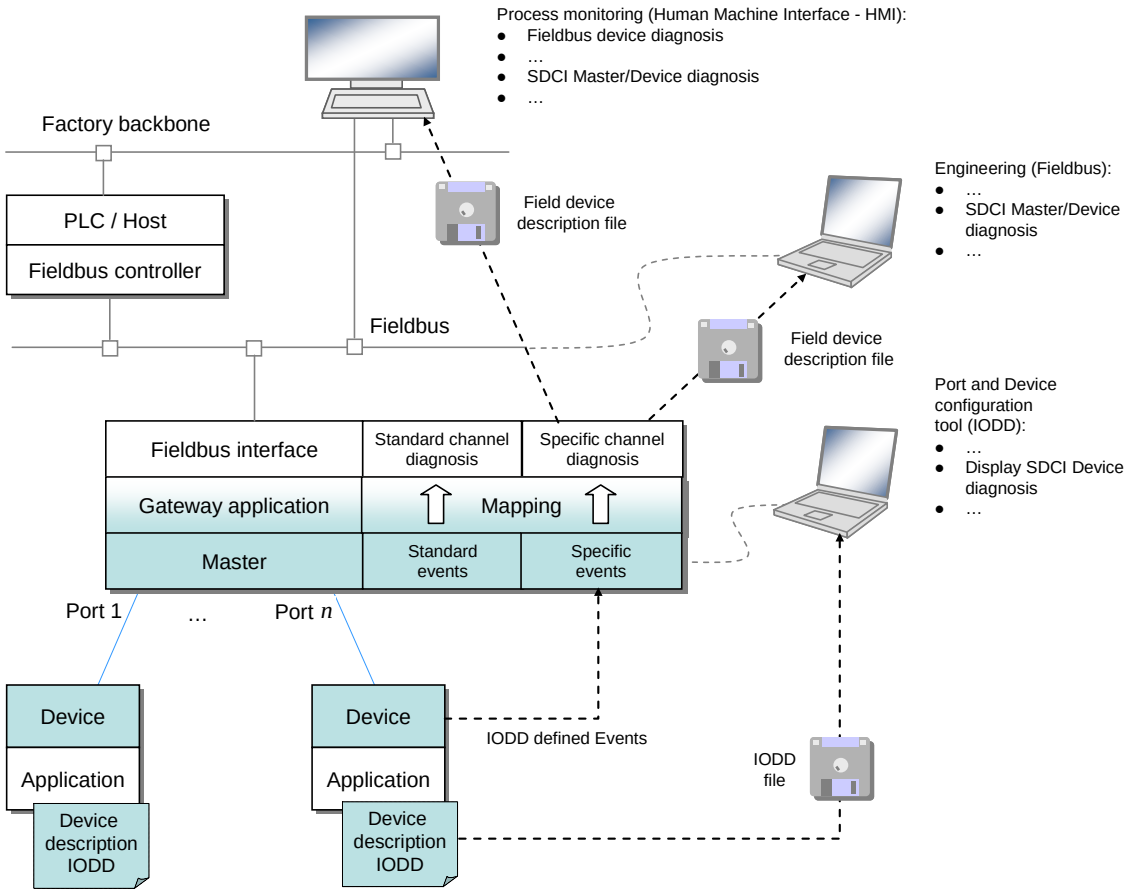
The gateway application is able to start or stop the Diagnosis Unit (see Figure 95). When stopped, the DU is defering any received AL_Event.ind call until the DU is started again.

The special DS_UPLOAD_REQ Event (see 10.4 and Table D.2) of a Device shall be redirected to the common Master application Data Storage. Those Events are acknowledged by the DU itself and not propagated to the gateway.

Figure 106 shows an example of the diagnosis information flow through a complete SDCI/fieldbus system.

NOTE The flow can end at the Master/PDCT or be more integrated depending on the fieldbus capabilities.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



NOTE Blue shaded areas indicate features specified in this standard.

Figure 106 – System overview of SDCI diagnosis information propagation via Events

11.6 PD Exchange (PDE)

11.6.1 General

The Process Data Exchange provides the transmission of Process Data between the gateway application and the connected Device.

After an established communication and Data Storage, the port is ready for any On-request Data (ODE) transfers. The Process Data communication is enabled whenever the specific port or all ports are switched to the OPERATE mode.

11.6.2 Process Data mapping

According to 11.2.2.4 the input and output Process Data are mapped to a specific part of the gateway Process Data stream.

Figure 107 shows a sample mapping of the Process Data from 3 Master ports to the Gateway Process Data stream.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

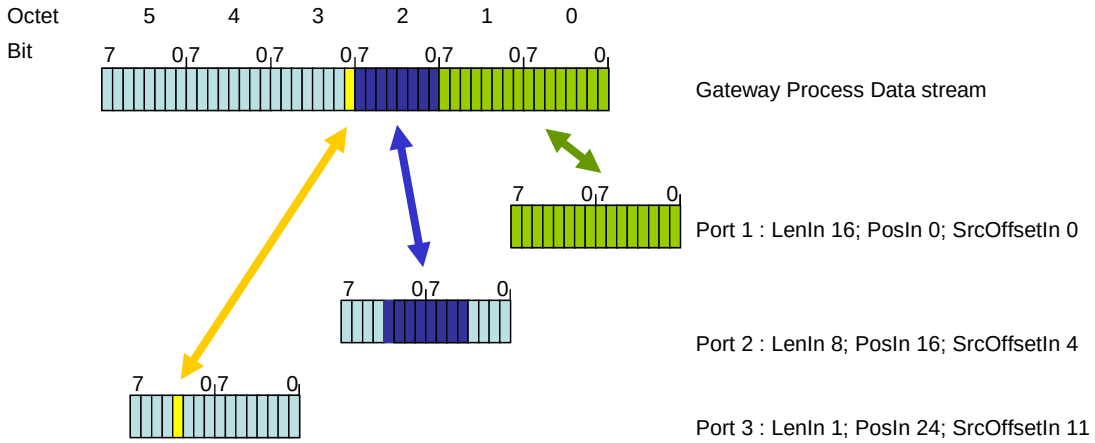


Figure 107 – Process Data mapping from ports to the gateway data stream

11.6.3 Process Data invalid/valid qualifier status

A sample transmission of an output PD qualifier status "invalid" from Master AL to Device AL is shown in the upper section of Figure 108.

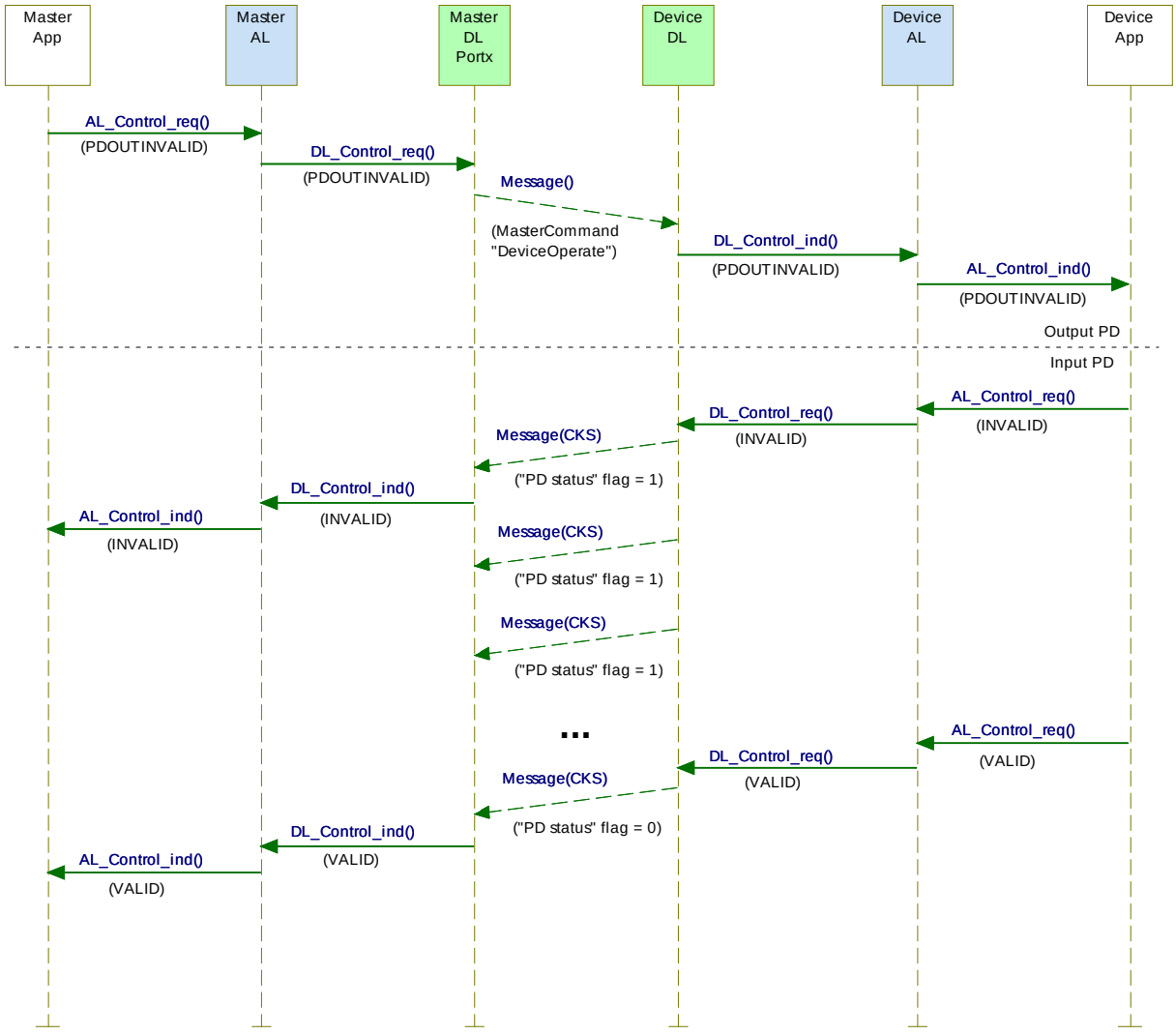


Figure 108 – Propagation of PD qualifier status between Master and Device

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The Master informs the Device about the output Process Data qualifier status "valid/invalid" by sending MasterCommands (see Table B.2) to the Direct Parameter page 1 (see 7.3.7.1).

For input Process Data the Device sends the Process Data qualifier status in every single message as the "PD status" flag in the Checksum / Status (CKS) octet (see A.1.5) of the Device message. A sample transmission of the input PD qualifier status "valid" from Device AL to Master AL is shown in the lower section of Figure 108.

Any perturbation while in interleave transmission mode leads to an input or output Process Data qualifier status "invalid" indication respectively.

11.7 Port and Device configuration tool (PDCT)

11.7.1 General

Figure 93 and Figure 106 demonstrate the necessity of a tool to configure ports, parameterize the Device, display diagnosis information, and provide identification and maintenance information. Depending on the degree of integration into a fieldbus system, the PDCT functions can be reduced, for example if the port configuration can be achieved via the field device description file of the particular fieldbus.

The PDCT functionality can be integrated partially (navigation, parameter transfer, etc.) or completely into the engineering tool of the particular fieldbus.

11.7.2 Basic layout examples

Figure 109 shows one example of a PDCT display layout.

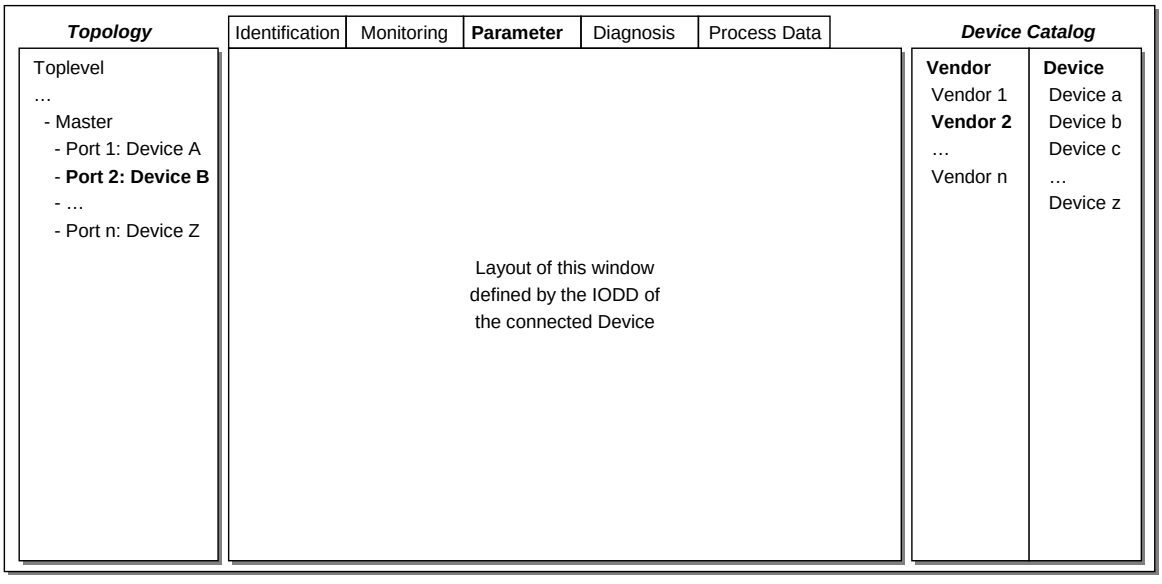


Figure 109 – Example 1 of a PDCT display layout

The PDCT display should always provide a navigation window for a project or a network topology, a window for the particular view on a chosen Device that is defined by its IODD, and a window for the available Devices based on the installed IODD files.

Figure 110 shows another example of a PDCT display layout.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

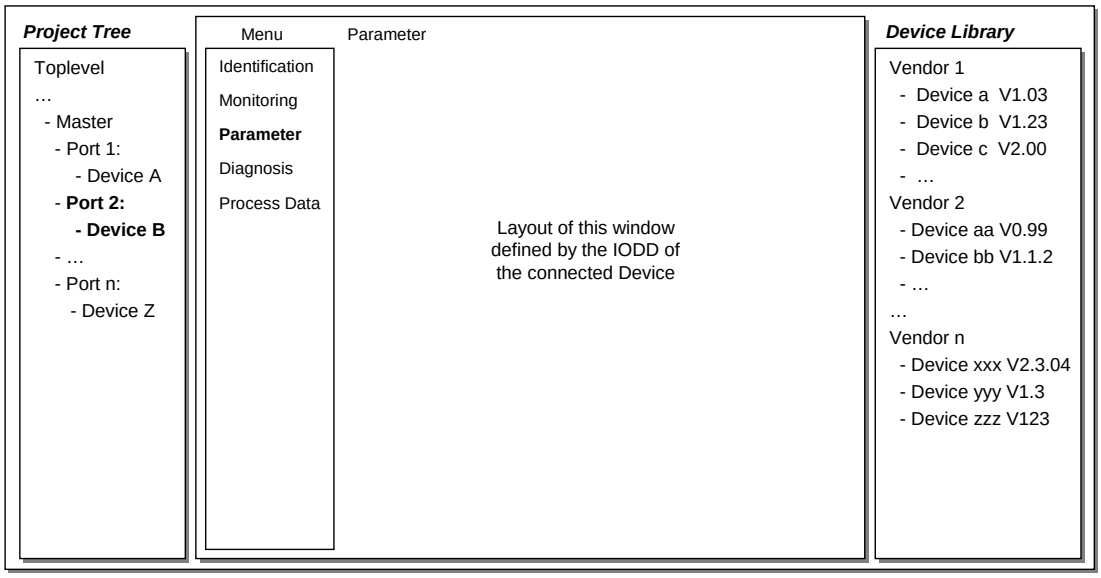


Figure 110 – Example 2 of a PDCT display layout

NOTE Further information can be retrieved from IEC/TR 62453-61.

11.8 Gateway application

11.8.1 General

The Gateway application depends on the individual host system (fieldbus, PLC, etc.) the Master applications are embedded in. It is the responsibility of the individual system to specify the mapping of the Master services and variables.

11.8.2 Changing Device configuration including Data Storage

After each change of Device configuration/parameterization (CVID and/or CDID, see 9.2.2.2), the associated previously stored data set within the Master shall be cleared or marked invalid via the variable DS_Delete.

11.8.3 Parameter server and recipe control

The Master may combine the entire parameter sets of the connected Devices together with all other relevant data for its own operation, and make this data available for higher level applications. For example, this data may be saved within a parameter server which may be accessed by a PLC program to change recipe parameters, thus supporting flexible manufacturing.

NOTE The structure of the data exchanged between the Master and the parameter server is outside the scope of this standard.

11.8.4 Anonymous parameters

An alternative to using a Port and Device Configuration Tool is necessary for some gateway interfaces. For these interfaces, it is recommended that the gateway interface allows the host to send it a block of 10 unnamed octets of Device configuration data for each Device attached to the Master. The gateway interface will then use the AL_Write service to deliver the octets for each Device to Direct Parameter page 2 of the associated Device.

NOTE Integration specifications are out of the scope of this standard.

This approach is shown in Figure 111.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

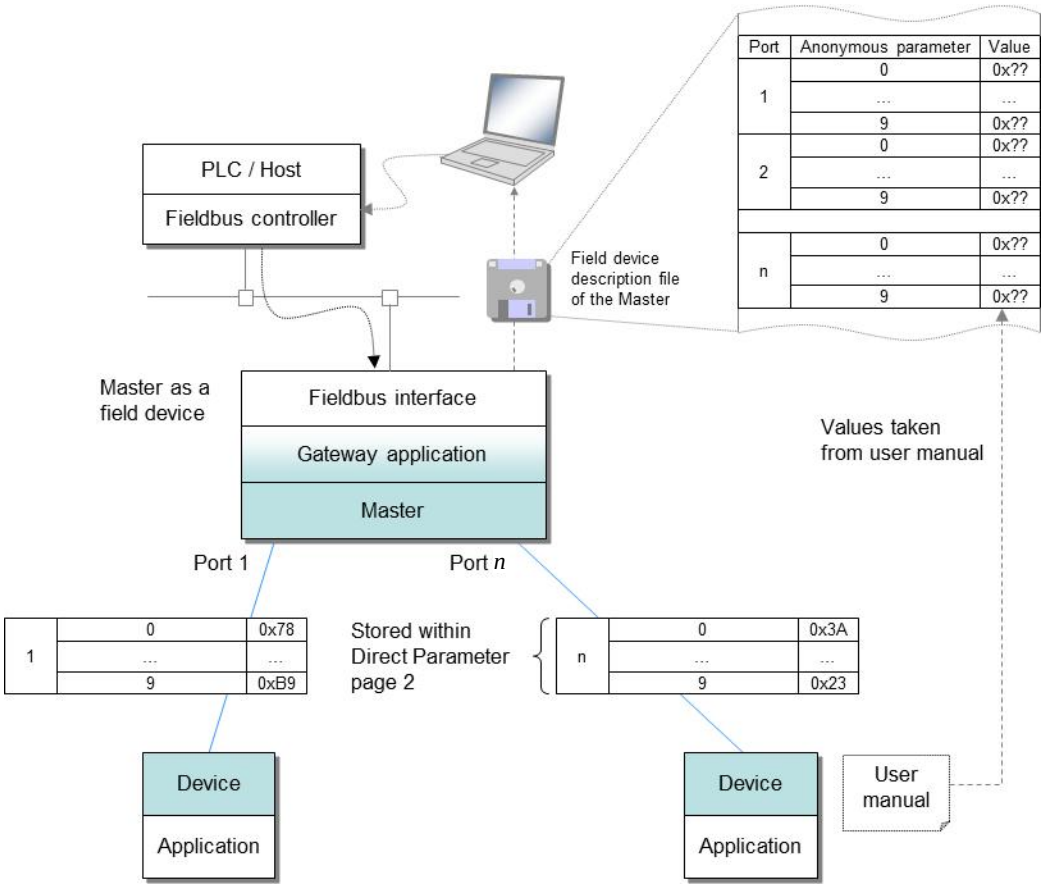


Figure 111 – Alternative Device configuration

11.8.5 Virtual port mode DIwithSDCI

This optional operational mode provides a possibility to use a Device with SIO capability in the DIwithSDCI mode and allow the higher level system (for example PLC) to exchange On-request Data acyclically. Preferably, this will take place when parameters are to be changed, at production stop, or diagnosis intervals.

This operational mode simplifies the control program due to the omission of configuration before and after an acyclic access.

In principle the gateway application realizes this operational mode virtually. It is solely in a position to decide within the individual states what the next steps can be.

The CM does not know this operational mode. The gateway application reads the configuration data hold by the CM and uses services from SM and AL to realize this operational mode.

The following rules shall be observed when implementing DIwithSDCI.

- The DI signal of the Device is not valid during the acyclic access of the gateway application.
- It is likely that an invalid DI signal is detected very late. Thus, only after the next acyclic access an Event "PDInvalid" can be raised and wire break or Device replacement can be detected.
- The access will consume more time due to establishing communication and fallback procedures including Data Storage.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- The InspectionLevel shall at least comprise TYPE_COMP in order to detect an illegal Device.

The state diagram in Figure 112 shows the individual states for the virtual operational mode DIwithSDCI.

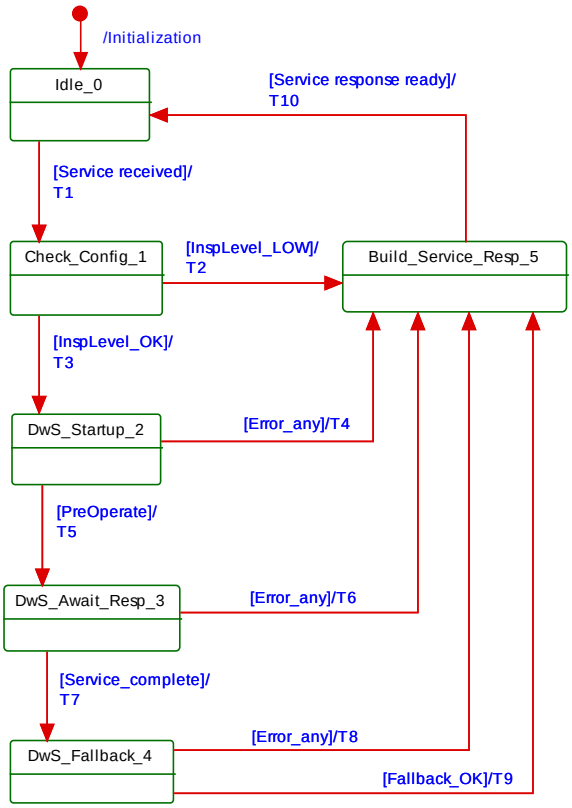


Figure 112 – Virtual port mode "DIwithSDCI"

Table 104 shows the states and transitions of the virtual port mode "DIwithSDCI".

Table 104 – State transitions of the state machine "DIwithSDCI"

STATE NAME	STATE DESCRIPTION
Idle_0	For the higher level control program the port is configured for the operational mode DIwithSDCI and waits on service requests.
Check_Config_1	Within this state the InspectionLevel of the port is checked for a sufficient level.
DwS_StartUp_2	Within this state a complete startup until the PREOPERATE state is performed. This can include Data Storage and Event handling.
DwS_Await_Resp_3	Wait on responses (AL-Read.rsp or AL-Write.rsp).
DwS_FallBack_4	After accomplishing the services, the Device is switched back to the SIO mode via the MasterCommand "Fallback" and the port will be configured as a DI.
Build_Service_Resp_5	The service response (positive or negative) will be created to finalize the service to the higher level control program.

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T1	0	1	-
T2	1	5	-
T3	1	2	Invoke SM_SetPortConfig (to COMx)
T4	2	5	Invoke SM_SetPortConfig (to DI)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	SOURCE STATE	TARGET STATE	ACTION
T5	2	3	Invoke Service (AL_Read.req oder AL_Write.req)
T6	3	5	Invoke SM_SetPortConfig (to DI)
T7	3	4	-
T8	4	5	Invoke SM_SetPortConfig (to DI)
T9	4	5	Invoke SM_SetPortConfig (to DI)
T10	5	0	Invoke corresponding AL service response

INTERNAL ITEMS	TYPE	DEFINITION
InspLevel_LOW	Bool	The necessary InspectionLevel is not configured in order to detect the correct Device.
InspLevel_OK	Bool	The necessary InspectionLevel is configured to detect the correct Device.
PreOperate	Bool	State PREOPERATE is established
Service_complete	Bool	The gateway application received AL_Read.rsp or AL_Write.rsp
Fallback_OK	Bool	Fallback has been accomplished successfully
Error_any	Bool	Any error will quit this state

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex A
(normative)

Codings, timing constraints, and errors

A.1 General structure and encoding of M-sequences

A.1.1 Overview

The general concept of M-sequences is outlined in 7.3.3.2. Subclauses A.1.2 to A.1.6 provide a detailed description of the individual elements of M-sequences.

A.1.2 M-sequence control (MC)

The Master indicates the manner the user data (see A.1.4) shall be transmitted in an M-sequence control octet. This indication includes the transmission direction (read or write), the communication channel, and the address (offset) of the data on the communication channel. The structure of the M-sequence control octet is shown in Figure A.1.

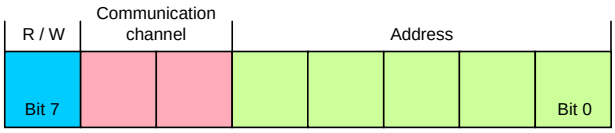


Figure A.1 – M-sequence control

Bit 0 to 4: Address

These bits indicate the address, i.e. the octet offset of the user data on the specified communication channel (see also Table A.1). In case of an ISDU channel, these bits are used for flow control of the ISDU data. The address, which means in this case the position of the user data within the ISDU, is only available indirectly (see 7.3.6.2).

Bit 5 to 6: Communication channel

These bits indicate the communication channel for the access to the user data. The defined values for the communication channel parameter are listed in Table A.1.

Table A.1 – Values of communication channel

Value	Definition
0	Process
1	Page
2	Diagnosis
3	ISDU

Bit 7: R/W

This bit indicates the transmission direction of the user data on the selected communication channel, i.e. read access (transmission of user data from Device to Master) or write access (transmission of user data from Master to Device). The defined values for the R/W parameter are listed in Table A.2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.2 – Values of R/W

Value	Definition
0	Write access
1	Read access

A Device is not required to support each and every of the 256 values of the M-sequence control octet. For read access to not implemented addresses or communication channels the value "0" shall be returned. A write access to not implemented addresses or communication channels shall be ignored.

A.1.3 Checksum / M-sequence type (CKT)

The M-sequence type is transmitted together with the checksum in the check/type octet. The structure of this octet is demonstrated in Figure A.2.

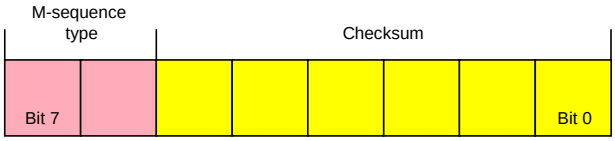


Figure A.2 – Checksum/M-sequence type octet

Bit 0 to 5: Checksum

These bits contain a 6 bit message checksum to ensure data integrity, see also A.1.6 and Clause H.1.

Bit 6 to 7: M-sequence type

These bits indicate the M-sequence type. Herewith, the Master specifies how the messages within the M-sequence are structured. Defined values for the M-sequence type parameter are listed in Table A.3.

Table A.3 – Values of M-sequence types

Value	Definition
0	Type 0
1	Type 1
2	Type 2 (see NOTE)
3	reserved
NOTE Subtypes depend on PD configuration and PD direction.	

A.1.4 User data (PD or OD)

User data is a general term for both Process Data and On-request Data. The length of user data can vary from 0 to 64 octets depending on M-sequence type and transmission direction (read/write). An overview of the available data types is shown in Table A.4. These data types can be arranged as records (different types) or arrays (same types).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.4 – Data types for user data

Data type	Reference
BooleanT	See E.2.2
UIntegerT	See E.2.3
IntegerT	See E.2.4
StringT	See E.2.6
OctetStringT	See E.2.7
Float32T	See E.2.5
TimeT	See E.2.8
TimeSpanT	See E.2.9

The detailed coding of the data types can be found in Annex E.

A.1.5 Checksum / status (CKS)

The checksum/status octet is part of the reply message from the Device to the Master. Its structure is shown in Figure A.3. It comprises a 6 bit checksum, a flag to indicate valid or invalid Process Data, and an Event flag.

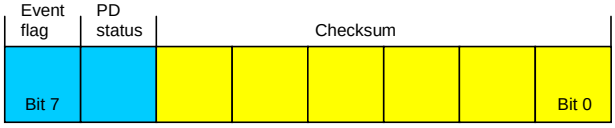


Figure A.3 – Checksum/status octet

Bit 0 to 5: Checksum

These bits contain a 6 bit checksum to ensure data integrity of the reply message. See also A.1.6 and H.1.

Bit 6: PD status

This bit indicates whether the Device can provide valid Process Data or not. Defined values for the parameter are listed in Table A.5.

This PD status flag shall be used for Devices with input Process Data. Devices with output Process Data shall always indicate "Process Data valid".

If the PD status flag is set to “Process Data invalid” within a message, all the input Process Data of the complete Process Data cycle are invalid.

Table A.5 – Values of PD status

Value	Definition
0	Process Data valid
1	Process Data invalid

Bit 7: Event flag

This bit indicates a Device initiative for the data category "Event" to be retrieved by the Master via the diagnosis communication channel (see Table A.1). The Device can report diagnosis information such as errors, warnings or notifications via Event response messages. Permissible values for the parameter are listed in Table A.6.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.6 – Values of the Event flag

Value	Definition
0	No Event
1	Event

A.1.6 Calculation of the checksum

The message checksum provides data integrity protection for data transmission from Master to Device and from Device to Master. Each UART data octet is protected by the UART parity bit (see Figure 18). Besides this individual data octet protection, all of the UART data octets in a message are XOR (exclusive or) processed octet by octet. The check/type octet is included with checksum bits set to "0". The resulting checksum octet is compressed from 8 to 6 bit in accordance with the conversion procedure in Figure A.4 and its associated formulas (see equations in (A.1)). The 6 bit compressed "Checksum6" is entered into the checksum/ M-sequence type octet (see Figure A.2). The same procedure takes place to secure the message from the Device to the Master. In this case the compressed checksum is entered into the checksum/status octet (see Figure A.3).

A seed value of 0x52 is used for the checksum calculation across the message. It is XORed with the first octet of the message (FC).

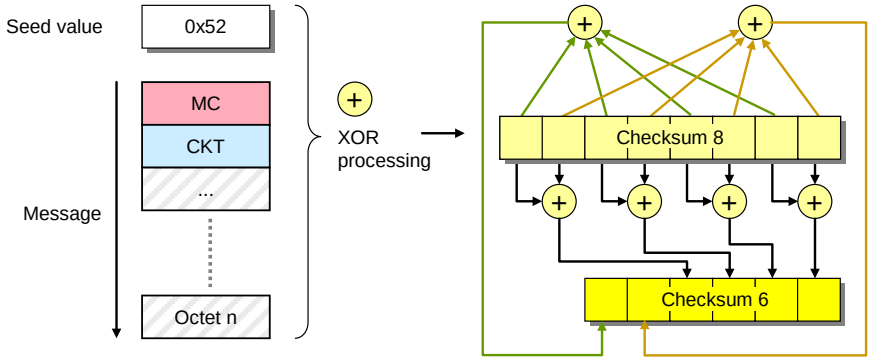


Figure A.4 – Principle of the checksum calculation and compression

The set of equations in (A.1) define the compression procedure from 8 to 6 bit in detail.

$$
\begin{aligned}
D5_6 &= D7_8 \text{ xor } D5_8 \text{ xor } D3_8 \text{ xor } D1_8 \\
D4_6 &= D6_8 \text{ xor } D4_8 \text{ xor } D2_8 \text{ xor } D0_8 \\
D3_6 &= D7_8 \text{ xor } D6_8 \\
D2_6 &= D5_8 \text{ xor } D4_8 \\
D1_6 &= D3_8 \text{ xor } D2_8 \\
D0_6 &= D1_8 \text{ xor } D0_8
\end{aligned}
\qquad \text{(A.1)}
$$

A.2 M-sequence types

A.2.1 Overview

Process Data and On-request Data use separate cyclic and acyclic communication channels (see Figure 7) to ensure scheduled and deterministic delivery of Process Data while delivery of On-request Data does not have consequences on the Process Data transmission performance.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Within SDCI, M-sequences provide the access to the communication channels via the M-sequence Control octet. The number of different M-sequence types meets the various requirements of sensors and actuators regarding their Process Data width. See Figure 37 for an overview of the available M-sequence types that are specified in A.2.2 to A.2.5. See A.2.6 for rules on how to use the M-sequence types.

A.2.2 M-sequence TYPE_0

M-sequence TYPE_0 is mandatory for all Devices.

M-sequence TYPE_0 only transmits On-request Data. One octet of user data is read or written per cycle. This M-sequence is shown in Figure A.5.

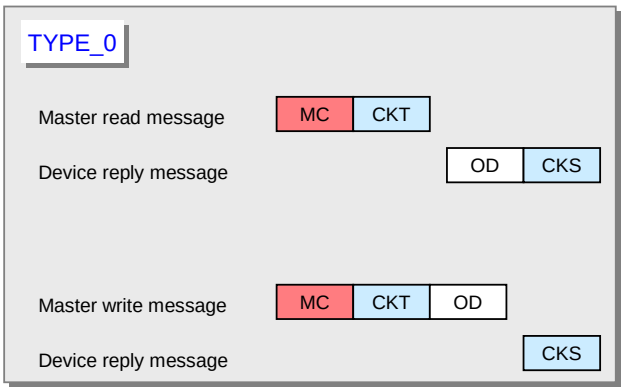


Figure A.5 – M-sequence TYPE_0

A.2.3 M-sequence TYPE_1_x

M-sequence TYPE_1_x is optional for all Devices.

M-sequence TYPE_1_1 is shown in Figure A.6.

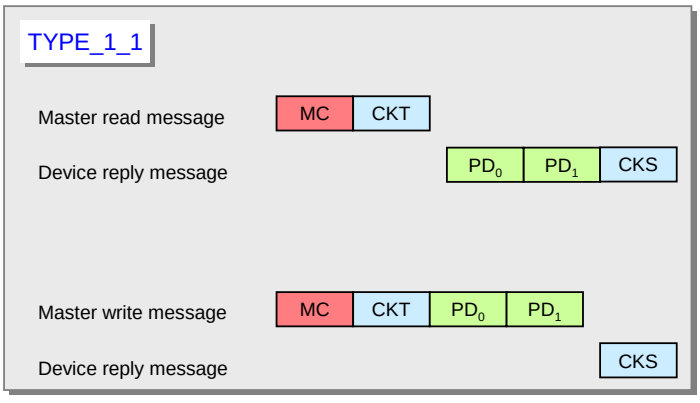


Figure A.6 – M-sequence TYPE_1_1

Two octets of Process Data are read or written per cycle. Address (bit offset) belongs to the process communication channel (see A.2.1).

In case of interleave mode (see 7.3.4.2) and odd-numbered PD length the remaining octets within the messages are padded with 0x00.

M-sequence TYPE_1_2 is shown in Figure A.7. Two octets of On-request Data are read or written per cycle.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

For write access to On-request Data via the page and diagnosis communication channels, only the first octet of On-request Data is evaluated. The Device shall ignore the remaining octets. The Master shall send all other ODs with "0x00".

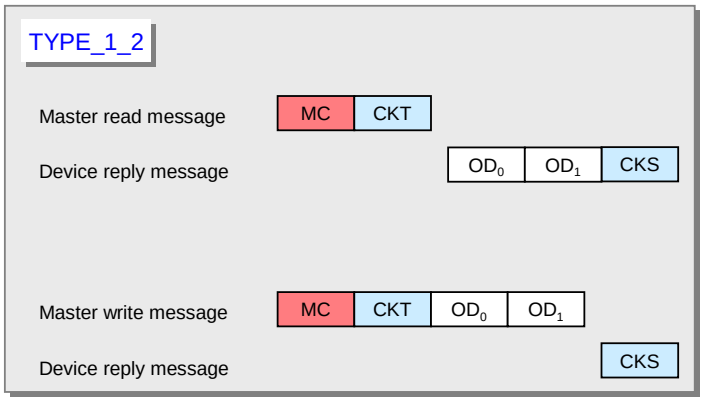


Figure A.7 – M-sequence TYPE_1_2

M-sequence TYPE_1_V providing variable (extendable) message length is shown in Figure A.8. A number of m octets of On-request Data are read or written per cycle.

For write access to On-request Data via the page and diagnosis communication channels, only the first octet (OD_0) of On-request Data is evaluated. The Device shall ignore the remaining octets. The Master shall send all other ODs with "0x00".

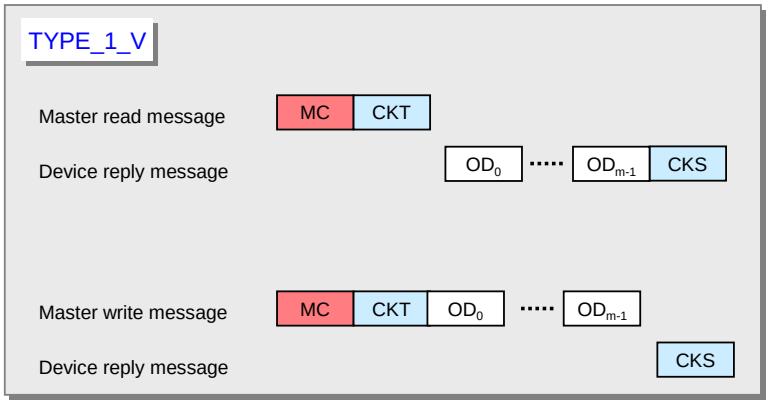


Figure A.8 – M-sequence TYPE_1_V

A.2.4 M-sequence TYPE_2_x

M-sequence TYPE_2_x is optional for all Devices. M-sequences TYPE_2_1 through TYPE_2_6 are defined. M-sequence TYPE_2_V provides variable (extendable) message length. M-sequence TYPE_2_x transmits Process Data and On-request Data in one message. The number of process and On-request Data read or written in each cycle depends on the type. The Address parameter (see Figure A.1) belongs in this case to the on-request communication channel. The Process Data address is specified implicitly starting at "0". The format of Process Data is characterizing the M-sequence TYPE_2_x.

M-sequence TYPE_2_1 transmits one octet of read Process Data and one octet of read or write On-request Data per cycle. This M-sequence type is shown in Figure A.9.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

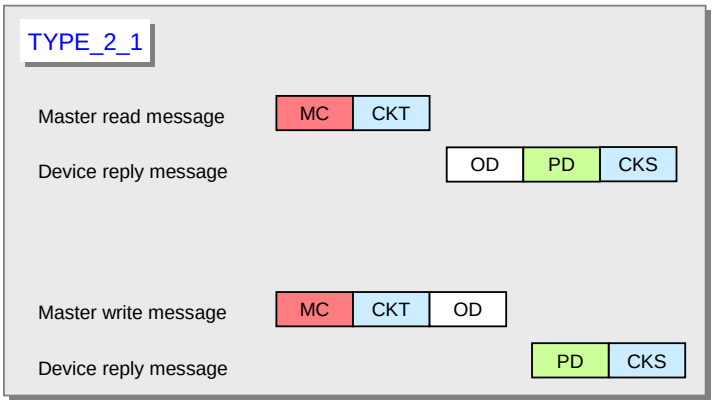


Figure A.9 – M-sequence TYPE_2_1

M-sequence TYPE_2_2 transmits 2 octets of read Process Data and one octet of On-request Data per cycle. This M-sequence type is shown in Figure A.10.

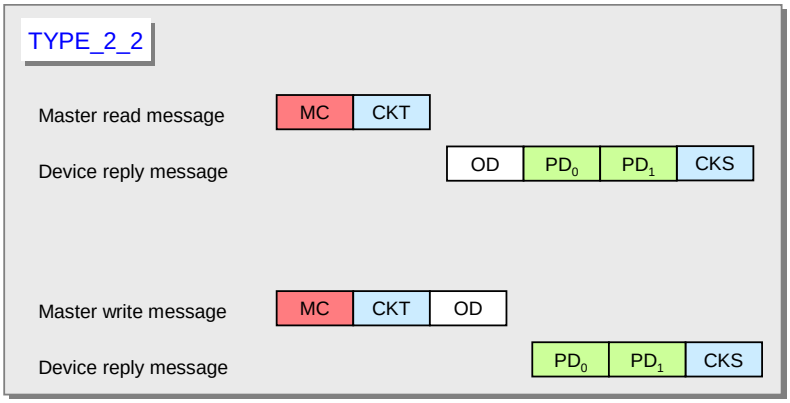


Figure A.10 – M-sequence TYPE_2_2

M-sequence TYPE_2_3 transmits one octet of write Process Data and one octet of read or write on-reqest data per cycle. This M-sequence type is shown in Figure A.11.

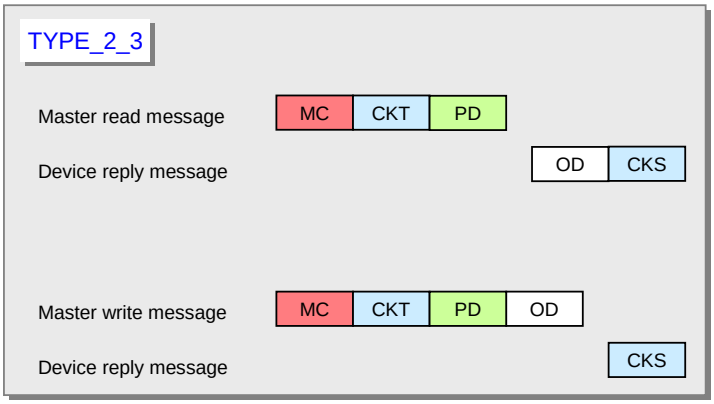


Figure A.11 – M-sequence TYPE_2_3

M-sequence TYPE_2_4 transmits 2 octets of write Process Data and one octet of read or write On-request Data per cycle. This M-sequence type is shown in Figure A.12

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

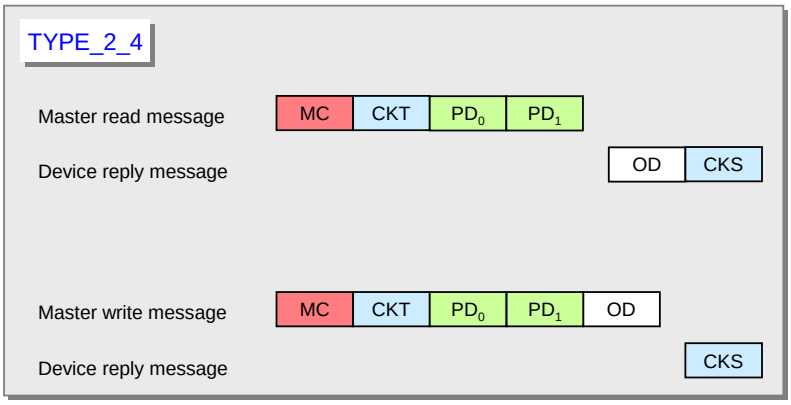


Figure A.12 – M-sequence TYPE_2_4

M-sequence TYPE_2_5 transmits one octet of write and read Process Data and one octet of read or write On-request Data per cycle. This M-sequence type is shown in Figure A.13.

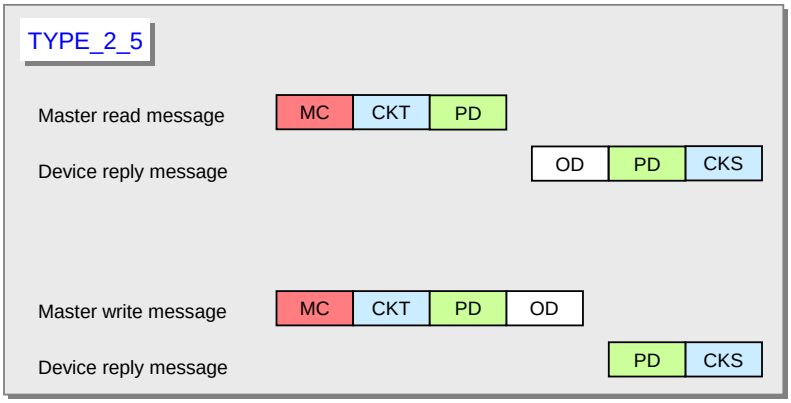


Figure A.13 – M-sequence TYPE_2_5

M-sequence TYPE_2_6 transmits 2 octets of write and read Process Data and one octet of read or write On-request Data per cycle. This M-sequence type is shown in Figure A.14.

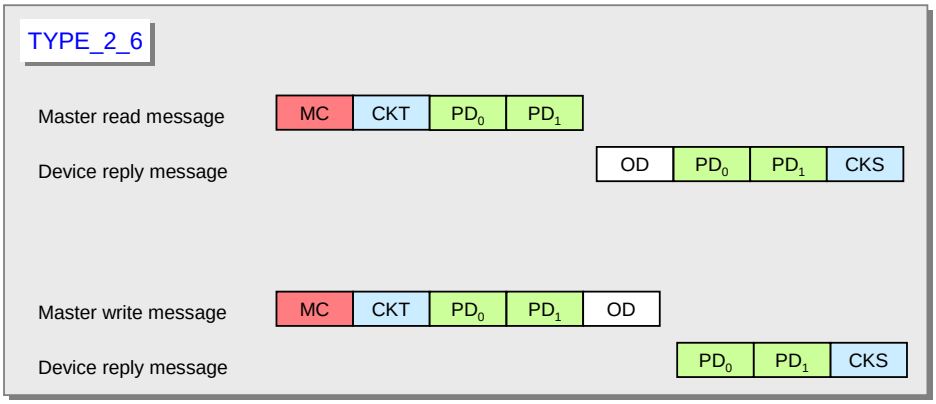


Figure A.14 – M-sequence TYPE_2_6

M-sequence TYPE_2_V transmits the entire write (read) ProcessDataIn n (k) octets per cycle. The range of n (k) is 0 to 32. Either PDin or PDout are not existing when n = 0 or k = 0. TYPE_2_V also transmits m octets of (segmented) read or write On-request Data per cycle using the address in Figure A.1. Permitted values for m are 1, 2, 8, and 32. This variable M-sequence type is shown in Figure A.15.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

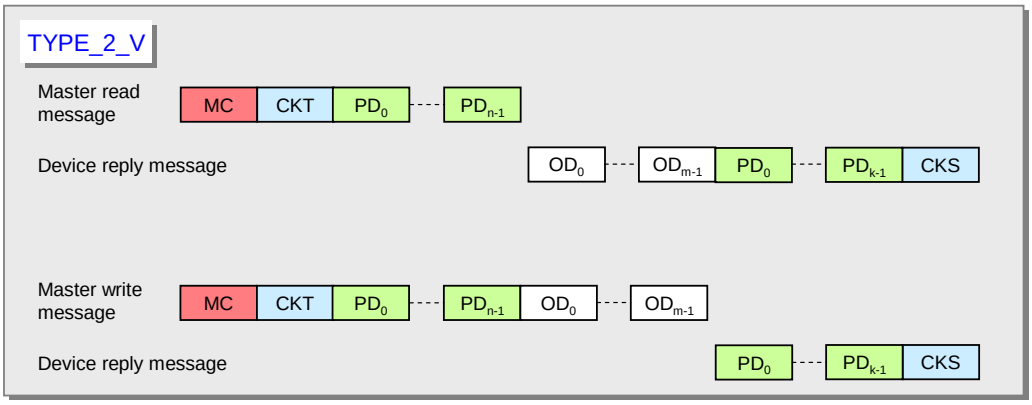


Figure A.15 – M-sequence TYPE_2_V

For write access to On-request Data via the page and diagnosis communication channels, only the first octet (OD_0) of On-request Data is evaluated. The Device shall ignore the remaining octets. The Master shall send all other ODs with "0".

A.2.5 M-sequence type 3

M-sequence type 3 is reserved and shall not be used.

A.2.6 M-sequence type usage for STARTUP, PREOPERATE and OPERATE modes

Table A.7 lists the M-sequence types for the STARTUP mode together with the minimum recovery time ($T_{initcyc}$) that shall be observed for Master implementations (see A.3.9). The M-sequence code refers to the coding in B.1.4.

Table A.7 – M-sequence types for the STARTUP mode

STARTUP M-sequence code	On-request Data Octets	M-sequence type	Minimum recovery time T_{BIT}
n/a	1	TYPE_0	100

Table A.8 lists the M-sequence types for the PREOPERATE mode together with the minimum recovery time ($T_{initcyc}$) that shall be observed for Master implementations.

Table A.8 – M-sequence types for the PREOPERATE mode

PREOPERATE M-sequence code	On-request Data Octets	M-sequence type	Minimum recovery time T_{BIT}
0	1	TYPE_0	100
1	2	TYPE_1_2	100
2	8	TYPE_1_V	210
3	32	TYPE_1_V	550
NOTE The minimum recovery time in PREOPERATE mode is a requirement for the Master.			

Table A.9 lists the M-sequence types for the OPERATE mode for legacy Devices. The minimum cycle time for Master in OPERATE mode is specified by the parameter "MinCycleTime" of the Device (see B.1.3).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.9 – M-sequence types for the OPERATE mode (legacy protocol)

OPERATE M-sequence code	On-request Data	Process Data (PD)		M-sequence type
	Octets	PDin	PDout	Legacy protocol (see [8])
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
don't care	2	3…32 octets	0…32 octets	TYPE_1_1/1_2 (interleaved)
don't care	2	0…32 octets	3…32 octets	TYPE_1_1/1_2 (interleaved)
don't care	1	1…8 bit	0	TYPE_2_1
don't care	1	9…16 bit	0	TYPE_2_2
don't care	1	0	1…8 bit	TYPE_2_3
don't care	1	0	9…16 bit	TYPE_2_4
don't care	1	1…8 bit	1…8 bit	TYPE_2_5

Table A.10 lists the M-sequence types for the OPERATE mode for Devices according to this standard. The minimum cycle time for Master in OPERATE mode is specified by the parameter MinCycleTime of the Device (see B.1.3).

Table A.10 – M-sequence types for the OPERATE mode

OPERATE M-sequence code	On-request Data	Process Data (PD)		M-sequence type
	Octets	PDin	PDout	
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
6	8	0	0	TYPE_1_V
7	32	0	0	TYPE_1_V
0	2	3…32 octets	0…32 octets	TYPE_1_1/1_2 interleaved
0	2	0…32 octets	3…32 octets	TYPE_1_1/1_2 interleaved
0	1	1…8 bit	0	TYPE_2_1
0	1	9…16 bit	0	TYPE_2_2
0	1	0	1…8 bit	TYPE_2_3
0	1	0	9…16 bit	TYPE_2_4
0	1	1…8 bit	1…8 bit	TYPE_2_5
0	1	9…16 bit	1…16 bit	TYPE_2_6
0	1	1…16 bit	9…16 bit	TYPE_2_6
4	1	0…32 octets	3…32 octets	TYPE_2_V
4	1	3…32 octets	0…32 octets	TYPE_2_V
5	2	>0 bit, octets	≥0 bit, octets	TYPE_2_V
5	2	≥0 bit, octets	>0 bit, octets	TYPE_2_V
6	8	>0 bit, octets	≥0 bit, octets	TYPE_2_V
6	8	≥0 bit, octets	>0 bit, octets	TYPE_2_V
7	32	>0 bit, octets	≥0 bit, octets	TYPE_2_V
7	32	≥0 bit, octets	>0 bit, octets	TYPE_2_V

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.3 Timing constraints

A.3.1 General

The interactions of a Master and its Device are characterized by several time constraints that apply to the UART frame, Master and Device message transmission times, supplemented by response, cycle, delay, and recovery times.

A.3.2 Bit time

The bit time T_{BIT} is the time it takes to transmit a single bit. It is the inverse value of the transmission rate (see Equation (A.2)).

$$T_{BIT} = 1/(\text{transmission rate}) \quad \text{(A.2)}$$

Values for T_{BIT} are specified in Table 8.

A.3.3 UART frame transmission delay of Master (ports)

The UART frame transmission delay t_1 of a port is the duration between the end of the stop bit of a UART frame and the beginning of the start bit of the next UART frame. The port shall transmit the UART frames within a maximum delay of one bit time (see Equation (A.3)).

$$0 \leq t_1 \leq 1\ T_{BIT} \quad \text{(A.3)}$$

A.3.4 UART frame transmission delay of Devices

The Device's UART frame transmission delay t_2 is the duration between the end of the stop bit of a UART frame and the beginning of the start bit of the next UART frame. The Device shall transmit the UART frames within a maximum delay of 3 bit times (see Equation (A.4)).

$$0 \leq t_2 \leq 3\ T_{BIT} \quad \text{(A.4)}$$

A.3.5 Response time of Devices

The Device's response time t_A is the duration between the end of the stop bit of a port's last UART frame being received and the beginning of the start bit of the first UART frame being sent. The Device shall observe a delay of at least one bit time but no more than 10 bit times (see Equation (A.5)).

$$1\ T_{BIT} \leq t_A \leq 10\ T_{BIT} \quad \text{(A.5)}$$

A.3.6 M-sequence time

Communication between a port and and its associated Device takes place in a fixed schedule, called the M-sequence time (see Equation (A.6)).

$$t_{\text{M-sequence}} = (m+n) \times 11 \times T_{BIT} + t_A + (m\text{-}1) \times t_1 + (n\text{-}1) \times t_2 \quad \text{(A.6)}$$

In this formula, m is the number of UART frames sent by the port to the Device and n is the number of UART frames sent by the Device to the port. The formula can only be used for estimates as the times t_1 and t_2 may not be constant.

Figure A.16 demonstrates the timings of an M-sequence consisting of a Master (port) message and a Device message.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

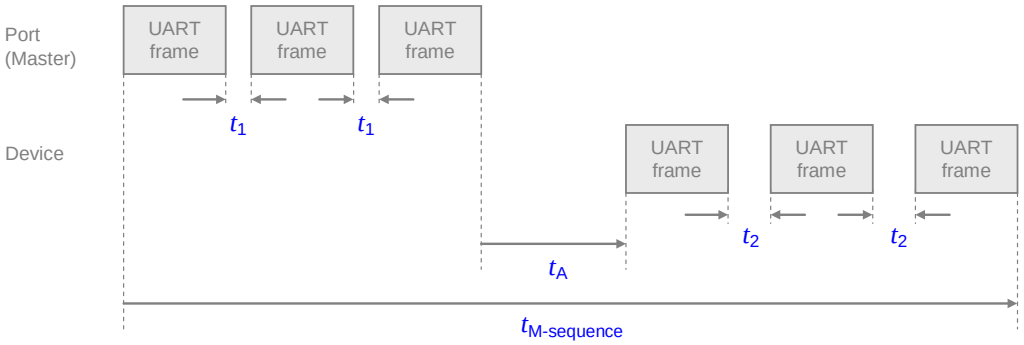


Figure A.16 – M-sequence timing

A.3.7 Cycle time

The cycle time t_{CYC} (see Equation (A.7)) depends on the Device's parameter “MinCycleTime” and the design and implementation of a Master and the number of ports.

$$t_{CYC} = t_{M\text{-sequence}} + t_{idle} \quad \text{(A.7)}$$

The adjustable Device parameter “MasterCycleTime” can be used for the design of a Device specific technology such as an actuator to derive the timing conditions for a default appropriate action such as de-activate or de-energize the actuator (see 7.3.3.5 "MaxCycleTime", 10.2, and 10.7.3).

Table A.11 lists recommended minimum cycle time values for the specified transmission mode of a port. The values are calculated based on M-sequence Type_2_1.

Table A.11 – Recommended MinCycleTimes

Transmission mode	t_{CYC}
COM1	18,0 ms
COM2	2,3 ms
COM3	0,4 ms

A.3.8 Idle time

The idle time t_{idle} results from the configured cycle time t_{CYC} and the M-sequence time $t_{M\text{-sequence}}$. With reference to a port, it comprises the time between the end of the message of a Device and the beginning of the next message from the Master (port).

The idle time shall be long enough for the Device to become ready to receive the next message.

A.3.9 Recovery time

The Master shall wait for a recovery time $t_{initcyc}$ between any two subsequent acyclic Device accesses while in the STARTUP or PREOPERATE phase (see A.2.6).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.4 Errors and remedies

A.4.1 UART errors

A.4.1.1 Parity errors

The UART parity bit (see Figure 18) and the checksum (see A.1.6) are two independent mechanisms to secure the data transfer. This means that for example two bit errors in different octets of a message, which are resulting in the correct checksum, can also be detected. Both mechanisms lead to the same error processing.

Remedy: The Master shall repeat the Master message 2 times (see 7.2.2.1). Devices shall reject all data with detected errors and create no reaction.

A.4.1.2 UART framing errors

The conditions for the correct detection of a UART frame are specified in 5.3.3.2. Error processing shall take place whenever perturbed signal shapes or incorrect timings lead to an invalid UART stop bit.

Remedy: See A.4.1.1.

A.4.2 Wake-up errors

The wake-up current pulse is specified in 5.3.3.3 and the wake-up procedures in 7.3.2.1. Several faults may occur during the attempts to establish communication.

Remedy: Retries are possible. See 7.3.2.1 for details.

A.4.3 Transmission errors

A.4.3.1 Checksum errors

The checksum mechanism is specified in A.1.6. Any checksum error leads to an error processing.

Remedy: See A.4.1.1.

A.4.3.2 Timeout errors

The diverse timing constraints with M-sequences are specified in A.3. Master (ports) and Devices are checking several critical timings such as lack of synchronism within messages.

Remedy: See A.4.1.1.

A.4.3.3 Collisions

A collision occurs whenever the Master and Device are sending simultaneously due to an error. This error is interpreted as a faulty M-sequence.

Remedy: See A.4.1.1.

A.4.4 Protocol errors

A protocol error occurs for example whenever the sequence of the segmented transmission of an ISDU is wrong (see flow control case in A.1.2).

Remedy: Abort of service with ErrorType information (see Annex C).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.5 General structure and encoding of ISDUs

A.5.1 Overview

The purpose and general structure of an ISDU is specified in 7.3.6.1. Subclauses A.5.2 to A.5.7 provide a detailed description of the individual elements of an ISDU and some examples.

A.5.2 I-Service

Figure A.17 shows the structure of the I-Service octet.



Figure A.17 – I-Service octet

Bits 0 to 3: Length
The encoding of the nibble Length of the ISDU is specified in Table A.14 .

Bits 4 to 7: I-Service
The encoding of the nibble I-Service of the ISDU is specified in Table A.12.

All other elements of the structure specified in 7.3.6.1 are transmitted as independent octets.

Table A.12 – Definition of the nibble "I-Service"

I-Service (binary)	Definition		Index format
	Master	Device	
0000	No Service	No Service	n/a
0001	Write Request	Reserved	8-bit Index
0010	Write Request	Reserved	8-bit Index and Subindex
0011	Write Request	Reserved	16-bit Index and Subindex
0100	Reserved	Write Response (-)	none
0101	Reserved	Write Response (+)	none
0110	Reserved	Reserved	
0111	Reserved	Reserved	
1000	Reserved	Reserved	
1001	Read Request	Reserved	8-bit Index
1010	Read Request	Reserved	8-bit Index and Subindex
1011	Read Request	Reserved	16-bit Index and Subindex
1100	Reserved	Read Response (-)	none
1101	Reserved	Read Response (+)	none
1110	Reserved	Reserved	
1111	Reserved	Reserved	

Table A.13 specifies the syntax of the ISDUs. ErrorType can be found in Annex C.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.13 – ISDU syntax

ISDU name	ISDU structure
Write Request	{I-Service(0x1), LEN, Index, [Data*], CHKPDU} ^ {I-Service(0x2), LEN, Index, Subindex, [Data*], CHKPDU} ^ {I-Service(0x3), LEN, Index, Index, Subindex, [Data*], CHKPDU}
Write Response (+)	I-Service(0x5), Length(0x2), CHKPDU
Write Response (-)	I-Service(0x4), Length(0x4), ErrorType, CHKPDU
Read Request	{I-Service(0x9), Length(0x3), Index, CHKPDU} ^ {I-Service(0xA), Length(0x4), Index, Subindex, CHKPDU} ^ {I-Service(0xB), Length(0x5), Index, Index, Subindex, CHKPDU}
Read Response (+)	I-Service(0xD), LEN, [Data*], CHKPDU
Read Response (-)	I-Service(0xC), Length(0x4), ErrorType, CHKPDU
Key LEN = {Length(0x1), ExtLength} ^ {Length}	

A.5.3 Extended length (ExtLength)

The number of octets transmitted in this I-Service, including all protocol information (6 octets), is specified in the “Length” element of an ISDU. If the total length is more than 15 octets, the length is specified using extended length information ("ExtLength"). Permissible values for “Length” and "ExtLength" are listed in Table A.14.

Table A.14 – Definition of nibble Length and octet ExtLength

I-Service	Length	ExtLength	Definition
0	0	n/a	No service, ISDU length is 1. Protocol use.
0	1	n/a	Device busy, ISDU length is 1. Protocol use.
0	2 to 15	n/a	Reserved and shall not be used
1 to 15	0	n/a	Reserved and shall not be used
1 to 15	1	0 to 16	Reserved and shall not be used
1 to 15	1	17 to 238	Length of ISDU in "ExtLength"
1 to 15	1	239 to 255	Reserved and shall not be used
1 to 15	2 to 15	n/a	Length of ISDU

A.5.4 Index and Subindex

The parameter address of the data object to be transmitted using the ISDU is specified in the “Index” element. “Index” has a range of values from 0 to 65 535 (see B.2.1 for constraints). Index values 0 and 1 shall be rejected by the Device.

There is no requirement for the Device to support all Index and Subindex values. The Device shall send a negative response to Index or Subindex values not supported.

The data element address of a structured parameter of the data object to be transmitted using the ISDU is specified in the “Subindex” element. “Subindex” has a range of values from 0 to 255, whereby a value of "0" is used to reference the entire data object (see Figure 5).

Table A.15 lists the Index formats used in the ISDU depending on the parameters transmitted.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.15 – Use of Index formats

Index	Subindex	Index format of ISDU
0 to 255	0	8 bit Index
0 to 255	1 to 255	8 bit Index and 8 bit Subindex
256 to 65 535	0 to 255	16 bit Index and 8 bit Subindex (see NOTE)
NOTE See B.2.1 for constraints on the Index range		

A.5.5 Data

The “Data” element can contain the data objects specified in Annex B or Device specific data objects respectively. The data length corresponds to the entries in the “Length” element minus the ISDU protocol elements.

A.5.6 Check ISDU (CHKPDU)

The “CHKPDU” element provides data integrity protection. The sender calculates the value of "CHKPDU" by XOR processing all of the octets of an ISDU, including "CHKPDU" with a preliminary value "0", which is then replaced by the result of the calculation (see Figure A.18).

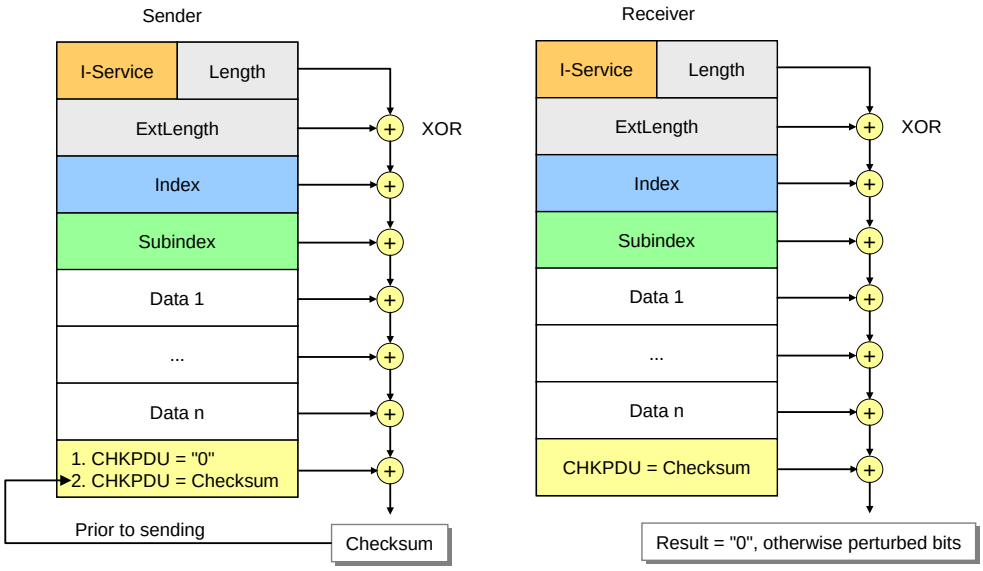


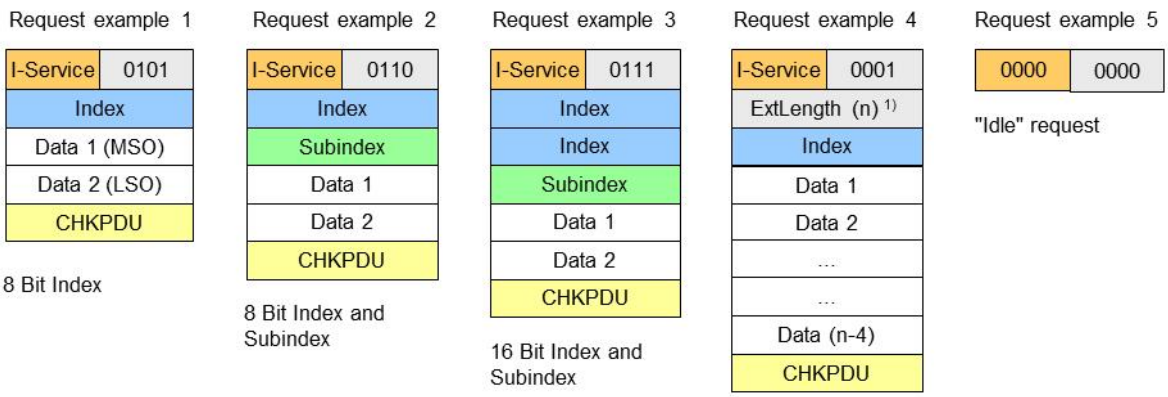
Figure A.18 – Check of ISDU integrity via CHKPDU

The receiver checks whether XOR processing of all of the octets of the ISDU will lead to the result "0" (see Figure A.18). If the result is different from "0", error processing shall take place. See also A.1.6.

A.5.7 ISDU examples

Figure A.19 demonstrates typical examples of request formats for ISDUs, which are explained in the following paragraphs.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



1) Overall ISDU ExtLength = *n* (1 to 238); Length = 1 ("0001")

Figure A.19 – Examples of request formats for ISDUs

The ISDU request in example 1 comprises one Index element allowing addressing from 0 to 254 (see Table A.15). In this example the Subindex is "0" and the whole content of Index is Data 1 with the most significant octet (MSO) and Data 2 with the least significant octet (LSO). The total length is 5 ("0101").

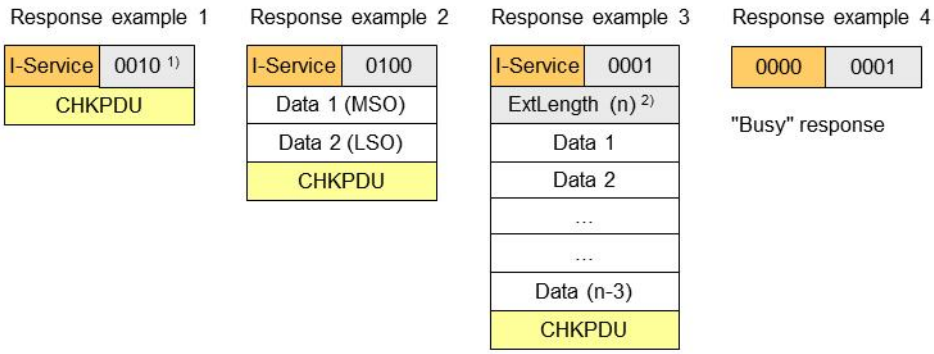
The ISDU request in example 2 comprises one Index element allowing addressing from 0 to 254 and the Subindex element allowing addressing an element of a data structure. The total length is 6 ("0110").

The ISDU request in example 3 comprises two Index elements allowing to address from 256 to 65 535 (see Table A.15) and the Subindex element allowing to address an element of a data structure. The total length is 7 ("0111").

The ISDU request in example 4 comprises one Index element and the ExtLength element indicating the number of ISDU elements (*n*), permitting numbers from 17 to 238. In this case the Length element has the value "1".

The ISDU request "Idle" in example 5 is used to indicate that no service is pending.

Figure A.20 demonstrates typical examples of response ISDUs, which are explained in the following paragraphs.



1) Minimum length = 2 ("0010")
2) Overall ISDU ExtLength = *n* (17 to 238);
Length = 1 ("0001")

Figure A.20 – Examples of response ISDUs

The ISDU response in example 1 shows the minimum value 2 for the Length element ("0010").

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The ISDU response in example 2 shows two Data elements and a total number of 4 elements in the Length element ("0100"). Data 1 carries the most significant octet (MSO) and Data 2 the least significant octet (LSO).

The ISDU response in example 3 shows the ExtLength element indicating the number of ISDU elements (*n*), permitting numbers from 17 to 238. In this case the Length element has the value "1".

The ISDU response "Busy" in example 4 is used when a Device is currently not able to respond to the read request of the Master due to the necessary preparation time for the response.

Figure A.21 shows a typical example of both a read and a write request ISDU, which are explained in the following paragraphs.

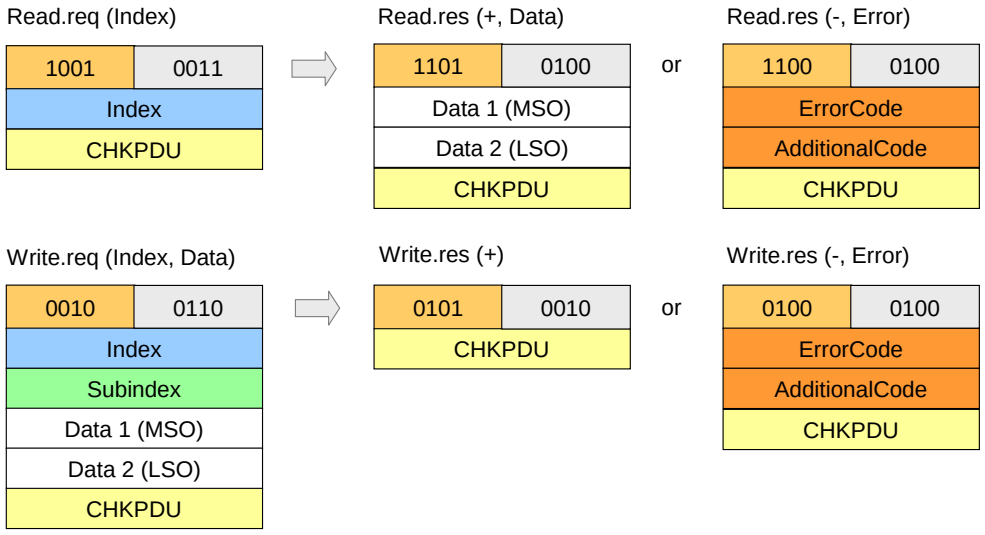


Figure A.21 – Examples of read and write request ISDUs

The code of the read request I-Service is "1001". According to Table A.13 this comprises an Index element. A successful read response (+) of the Device with code "1101" is shown next to the request with two Data elements. Total length is 4 ("0100"). An unsuccessful read response (-) of the Device with code "1100" is shown next in line. It carries the ErrorType with the two Data elements ErrorCode and AdditionalCode (see Annex C).

The code of the write request I-Service is "0010". According to Table A.13 this comprises an Index and a Subindex element. A successful write response (+) of the Device with code "0101" is shown next to the request with no Data elements. Total length is 2 ("0010"). An unsuccessful read response (-) of the Device with code "0100" is shown next in line. It carries the ErrorType with the two Data elements ErrorCode and AdditionalCode (see Annex C).

A.6 General structure and encoding of Events

A.6.1 General

In 7.3.8.1 and Table 56 the purpose and general structure of the Event memory is specified. This memory accommodates a StatusCode, several EventQualifiers and their associated EventCodes. The coding of these memory elements is specified in A.6.2 to A.6.5.

A.6.2 StatusCode type 1 (no details)

Figure A.22 shows the structure of this StatusCode.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE 1 StatusCode type 1 is only used in Events generated by legacy devices (see 7.3.8.1).

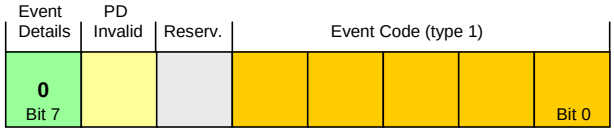


Figure A.22 – Structure of StatusCode type 1

Bits 0 to 4: EventCode (type 1)
The coding of this data structure is listed in Table A.16. The EventCodes are mapped into EventCodes (type 2) as listed in Annex D. See 7.3.8.2 for additional information.

Table A.16 – Mapping of EventCodes (type 1)

EventCode (type 1)	EventCode (type2)	Instance	Type	Mode
****1	0xFF80	Application	Notification	Event single shot
***1*	0xFF80	Application	Warning	Event single shot
1	0x6320	Application	Error	Event single shot
*1***	0xFF80	Application	Error	Event single shot
1****	0xFF10	Application	Error	Event single shot
Key * Do not care type 2 See Table D.1 and Table D.2				

Bit 5: Reserved
This bit is reserved and shall be set to zero in StatusCode type 1.

Bit 6: Reserved

NOTE 2 This bit is used in legacy protocol (see [8]) for PDinvalid indication.

Bit 7: Event Details
This bit indicates that no detailed Event information is available. It shall always be set to zero in StatusCode type 1.

A.6.3 StatusCode type 2 (with details)

Figure A.23 shows the structure of the StatusCode type 2.

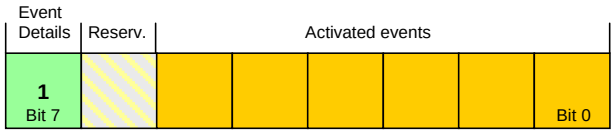


Figure A.23 – Structure of StatusCode type 2

Bits 0 to 5: Activated Events
Each bit is linked to an Event in the memory (see 7.3.8.1) as demonstrated in Figure A.24. Bit 0 is linked to Event 1, bit 1 to Event 2, etc. A bit with value "1" indicates that the corresponding EventQualifier and the EventCode have been entered in valid formats in the memory. A bit with value "0" indicates an invalid entry.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

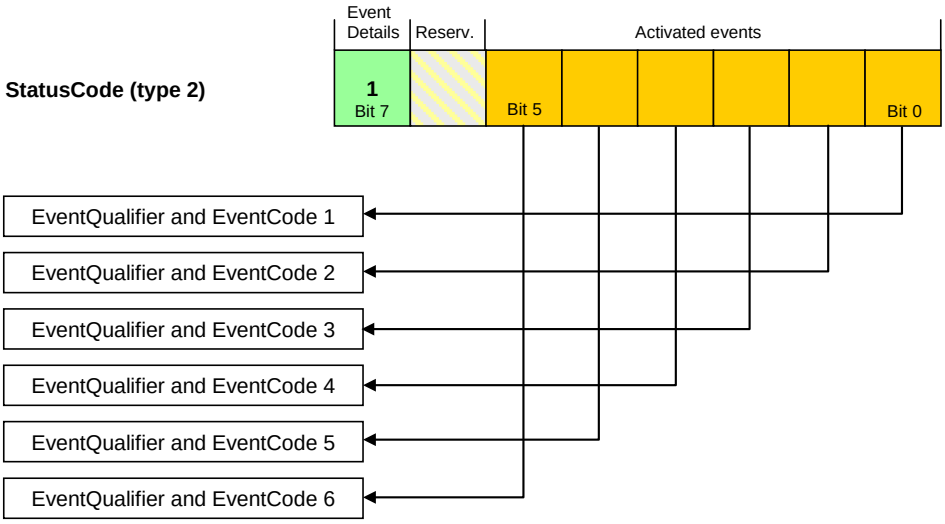


Figure A.24 – Indication of activated Events

Bit 6: Reserved
This bit is reserved and shall be set to zero.

NOTE This bit is used in the legacy protocol version according to [8] for PDinvalid indication.

Bit 7: Event Details
This bit indicates that detailed Event information is available. It shall always be set in StatusCode type 2.

A.6.4 EventQualifier

The structure of the EventQualifier is shown in Figure A.25.

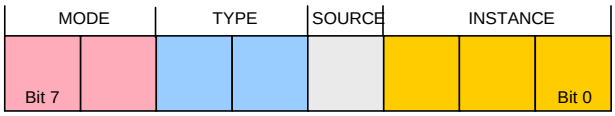


Figure A.25 – Structure of the EventQualifier

Bits 0 to 2: INSTANCE
These bits indicate the particular source (instance) of an Event thus refining its evaluation on the receiver side. Permissible values for INSTANCE are listed in Table A.17.

Table A.17 – Values of INSTANCE

Value	Definition
0	Unknown
1 to 3	Reserved
4	Application
5 to 7	Reserved

Bit 3: SOURCE
This bit indicates the source of the Event. Permissible values for SOURCE are listed in Table A.18.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table A.18 – Values of SOURCE

Value	Definition
0	Device (remote)
1	Master (local)

Bits 4 to 5: TYPE
These bits indicate the Event category. Permissible values for TYPE are listed in Table A.19.

Table A.19 – Values of TYPE

Value	Definition
0	Reserved
1	Notification
2	Warning
3	Error

Bits 6 to 7: MODE
These bits indicate the Event mode. Permissible values for MODE are listed in Table A.20.

Table A.20 – Values of MODE

Value	Definition
0	reserved
1	Event single shot
2	Event disappears
3	Event appears

A.6.5 EventCode

The EventCode entry contains the identifier of an actual Event. Permissible values for EventCode are listed in Annex D.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex B
(normative)

Parameter and commands

B.1 Direct Parameter page 1 and 2

B.1.1 Overview

In principle, the designer of a Device has a large amount of space for parameters and commands as shown in Figure 5. However, small sensors with a limited number of parameters and limited resources are striving for a simple subset. SDCI offers the so-called Direct Parameter pages 1 and 2 with a simplified access method (page communication channel according to Table A.1) to meet this requirement.

The range of Direct Parameters is structured as shown in Figure B.1. It is split into page 1 and page 2.

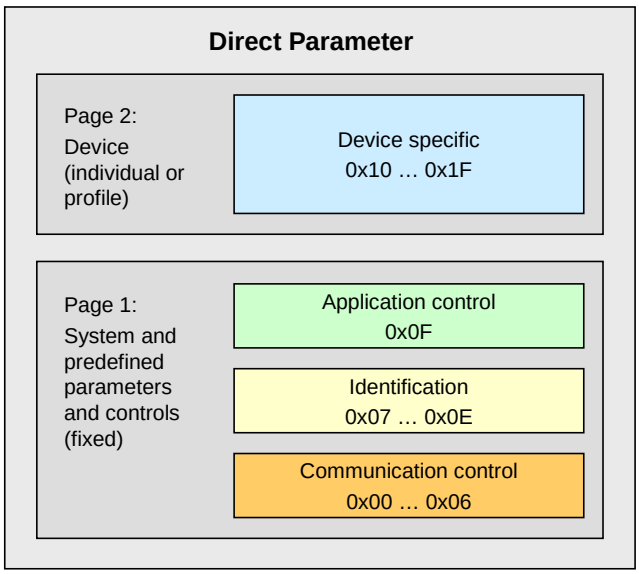


Figure B.1 – Classification and mapping of Direct Parameters

Page 1 ranges from 0x00 to 0x0F. It comprises the following categories of parameters:

- communication control;
- identification parameter;
- application control.

The Master application layer (AL) provides read only access to Direct Parameter page 1 as data objects (see 8.2.1) via Index 0. Single octets can be read via Index 0 and the corresponding Subindex. Subindex 1 indicates address 0x00 and Subindex 16 address 0x0F.

Page 2 ranges from 0x10 to 0x1F. This page comprises parameters optionally used by the individual Device technology. The Master application layer (AL) provides read/write access to Direct Parameter page 2 in form of data objects (see 8.2.1) via Index 1. Single octets can be written or read via Index 1 and the corresponding Subindex. Subindex 1 indicates address 0x10 and Subindex 16 address 0x1F.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A Device shall always return the value "0" upon a read access to Direct Parameter addresses, which are not implemented (for example in case of reserved parameter addresses or not supported optional parameters). The Device shall ignore a write access to not implemented parameters.

The structure of the Direct Parameter pages 1 and 2 is specified in Table B.1.

Table B.1 – Direct Parameter page 1 and 2

Address	Parameter name	Access	Implementation /reference	Description
Direct Parameter page 1				
0x00	Master-Command	W	Mandatory/ see B.1.2	Master command to switch to operating states (see NOTE 1)
0x01	MasterCycle-Time	R/W	Mandatory/ see B.1.3	Actual cycle duration used by the Master to address the Device. Can be used as a parameter to monitor Process Data transfer.
0x02	MinCycleTime	R	Mandatory/ see B.1.3	Minimum cycle duration supported by a Device. This is a performance feature of the Device and depends on its technology and implementation.
0x03	M-sequence Capability	R	Mandatory/ see B.1.4	Information about implemented options related to M-sequences and physical configuration
0x04	RevisionID	R/W	Mandatory/ see B.1.5	ID of the used protocol version for implementation (shall be set to 0x11)
0x05	ProcessDataIn	R	Mandatory/ see B.1.6	Number and structure of input data (Process Data from Device to Master)
0x06	ProcessData-Out	R	Mandatory/ see B.1.7	Number and structure of output data (Process Data from Master to Device)
0x07	VendorID 1 (MSB)	R	Mandatory/ see B.1.8	Unique vendor identification (see NOTE 2)
0x08	VendorID 2 (LSB)			
0x09	DeviceID 1 (Octet 2,MSB)	R/W	Mandatory/ see B.1.9	Unique Device identification allocated by a vendor
0x0A	DeviceID 2 (Octet 1)			
0x0B	DeviceID 3 (Ocet 0, LSB)			
0x0C	FunctionID 1 (MSB)	R	see B.1.10	Reserved (Engineering shall set both octets to "0x00")
0x0D	FunctionID 2 (LSB)			
0x0E		R	reserved	
0x0F	System-Command	W	Optional/ see B.1.11	Command interface for end user applications only and Devices without ISDU support (see NOTE 1)
Direct Parameter page 2				
0x10... 0x1F	Vendor specific	Optional	Optional/ see B.1.12	Device specific parameters
NOTE 1 A read operation returns unspecified values. NOTE 2 VendorIDs are assigned by the IO-Link consortium.				

B.1.2 MasterCommand

The Master application is able to check the status of a Device or to control its behaviour with the help of MasterCommands (see 7.3.7).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Permissible values for these parameters are specified in Table B.2.

Table B.2 – Types of MasterCommands

Value	MasterCommand	Description
0x00 to 0x59	Reserved	
0x5A	Fallback	Transition from communication to SIO mode. The Device shall execute this transition after 3 Master-CycleTimes and before 500 ms elapsed after the MasterCommand.
0x5B to 0x94	Reserved	
0x95	MasterIdent	Indicates a Master revision higher than 1.0
0x96	DeviceIdent	Start check of Direct Parameter page for changed entries
0x97	DeviceStartup	Switches the Device from OPERATE or PREOPERATE to STARTUP
0x98	ProcessDataOutputOperate	Process output data valid
0x99	DeviceOperate	Process output data invalid or not available. Switches the Device from STARTUP or PREOPERATE to OPERATE
0x9A	DevicePreoperate	Switches the Device from STARTUP to state PREOPERATE
0x9B to 0xFF	Reserved	

B.1.3 MasterCycleTime and MinCycleTime

The MasterCycleTime is a Master parameter and sets up the actual cycle time of a particular port.

The MinCycleTime is a Device parameter to inform the Master about the shortest cycle time supported by this Device.

See A.3.7 for the application of the MasterCycleTime and the MinCycleTime. The structure of these two parameters is shown in Figure B.2.

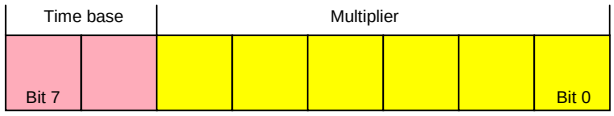


Figure B.2 – MinCycleTime

Bits 0 to 5: Multiplier

These bits contain a 6-bit multiplier for the calculation of MasterCycleTime or MinCycleTime. Permissible values for the multiplier are 0 to 63.

Bits 6 to 7: Time Base

These bits specify the time base for the calculation of MasterCycleTime or MinCycleTime.

When all bits are zero, (binary code 0x00), the Device has no MinCycleTime. In this case the Master shall use the calculated worst case M-sequence timing, that is with the M-sequence type used by the Device, and the maximum times for t_A and t_2 (see A.3.4 to A.3.6).

The permissible combinations for time base and multiplier are listed in Table B.3 along with the resulting values for MasterCycleTime or MinCycleTime.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table B.3 – Possible values of MasterCycleTime and MinCycleTime

Time base encoding	Time Base value	Calculation	Cycle Time
00	0,1 ms	Multiplier × Time Base	0,4 ms to 6,3 ms
01	0,4 ms	6,4 ms + Multiplier × Time Base	6,4 ms to 31,6 ms
10	1,6 ms	32,0 ms + Multiplier × Time Base	32,0 ms to 132,8 ms
11	Reserved	Reserved	Reserved
NOTE The value 0,4 results from the minimum possible transmission time according to A.3.7.			

B.1.4 M-sequence Capability

The structure of the M-sequence Capability parameter is shown in Figure B.3.

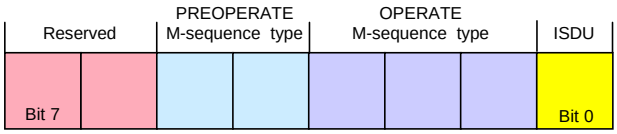


Figure B.3 – M-sequence Capability

Bit 0: ISDU

This bit indicates whether or not the ISDU communication channel is supported. Permissible values for ISDU are listed in Table B.4.

Table B.4 – Values of ISDU

Value	Definition
0	ISDU not supported
1	ISDU supported

Bits 1 to 3: Coding of the OPERATE M-sequence type

This parameter indicates the available M-sequence type during the OPERATE state. Permissible codes for the OPERATE M-sequence type are listed in Table A.9 for legacy Devices and in Table A.10 for Devices according to this standard.

Bits 4 to 5: Coding of the PREOPERATE M-sequence type

This parameter indicates the available M-sequence type during the PREOPERATE state. Permissible codes for the PREOPERATE M-sequence type are listed in Table A.8.

Bits 6 to 7: Reserved

These bits are reserved and shall be set to zero in this version of the specification.

B.1.5 RevisionID (RID)

The RevisionID parameter is the two-digit version number of the SDCI protocol currently used within the Device. Its structure is shown in Figure B.4. The initial value of RevisionID at powerup is the inherent value for protocol RevisionID. It can be overwritten (see 10.6.3) until the next powerup.

This revision of the standard specifies protocol version 1.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE The legacy protocol version 1.0 is specified in [8].



Figure B.4 – RevisionID

Bits 0 to 3: MinorRev

These bits contain the minor digit of the version number, for example 0 for the protocol version 1.0. Permissible values for MinorRev are 0x0 to 0xF.

Bits 4 to 7: MajorRev

These bits contain the major digit of the version number, for example 1 for the protocol version 1.0. Permissible values for MajorRev are 0x0 to 0xF.

B.1.6 ProcessDataIn

The structure of the ProcessDataIn parameter is shown in Figure B.5.

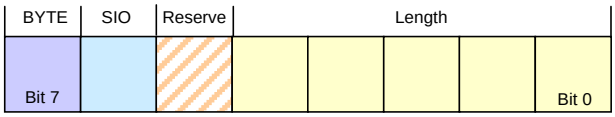


Figure B.5 – ProcessDataIn

Bits 0 to 4: Length

These bits contain the length of the input data (Process Data from Device to Master) in the length unit designated in the BYTE parameter bit. Permissible codes for Length are specified in Table B.6.

Bit 5: Reserve

This bit is reserved and shall be set to zero in this version of the specification.

Bit 6: SIO

This bit indicates whether the Device provides a switching signal in SIO mode. Permissible values for SIO are listed in Table B.5.

Table B.5 – Values of SIO

Value	Definition
0	SIO mode not supported
1	SIO mode supported

Bit 7: BYTE

This bit indicates the length unit for Length. Permissible values for BYTE and the resulting definition of the Process Data length in conjunction with Length are listed in Table B.6.

Table B.6 – Permitted combinations of BYTE and Length

BYTE	Length	Definition
0	0	no Process Data
0	1	1 bit Process Data, structured in bits
0	*n* (2-15)	*n* bit Process Data, structured in bits

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

BYTE	Length	Definition
0	16	16 bit Process Data, structured in bits
0	17 to 31	Reserved
1	0, 1	Reserved
1	2	3 octets Process Data, structured in octets
1	*n* (3-30)	*n*+1 octets Process Data, structured in octets
1	31	32 octets Process Data, structured in octets

B.1.7 ProcessDataOut

The structure of the ProcessDataOut parameter is the same as with ProcessDataIn, except with bit 6 ("SIO") reserved.

B.1.8 VendorID (VID)

These octets contain a worldwide unique value per vendor.

NOTE VendorIDs are assigned by the IO-Link consortium.

B.1.9 DeviceID (DID)

These octets contain the currently used DeviceID. A value of "0" is not permitted. The initial value of DeviceID at powerup is the inherent value of DeviceID. It can be overwritten (see 10.6.2) until the next powerup.

NOTE The communication parameters MinCycleTime, M-sequence Capability, Process Data In and Process Data Out can be changed to achieve compatibility to the requested DeviceID.

B.1.10 FunctionID (FID)

This parameter will be defined in a later version.

B.1.11 SystemCommand

Only Devices without ISDU support shall use the parameter SystemCommand in the Direct Parameter page 1. The implementation of SystemCommand is optional. See Table B.9 for a detailed description of the SystemCommand functions.

NOTE The SystemCommand on the Direct Parameter page 1 does not provide a positive or negative response upon execution of a selected function.

B.1.12 Device specific Direct Parameter page 2

The Device specific Direct Parameters are a set of parameters available to the Device specific technology. The implementation of Device specific Direct Parameters is optional.

NOTE The complete parameter list of the Direct Parameter page 2 is read or write accessible via index 1 (see B.1.1).

B.2 Predefined Device parameters

B.2.1 Overview

The many different technologies and designs of sensors and actuators require individual and easy access to complex parameters and commands beyond the capabilities of the Direct Parameter page 2. From a Master's point of view, these complex parameters and commands are called application data objects. So-called ISDU "containers" are the transfer means to exchange application data objects or short data objects. The index of the ISDU is used to

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

address the data objects. Figure B.6 shows the general mapping of data objects for the ISDU transmission.

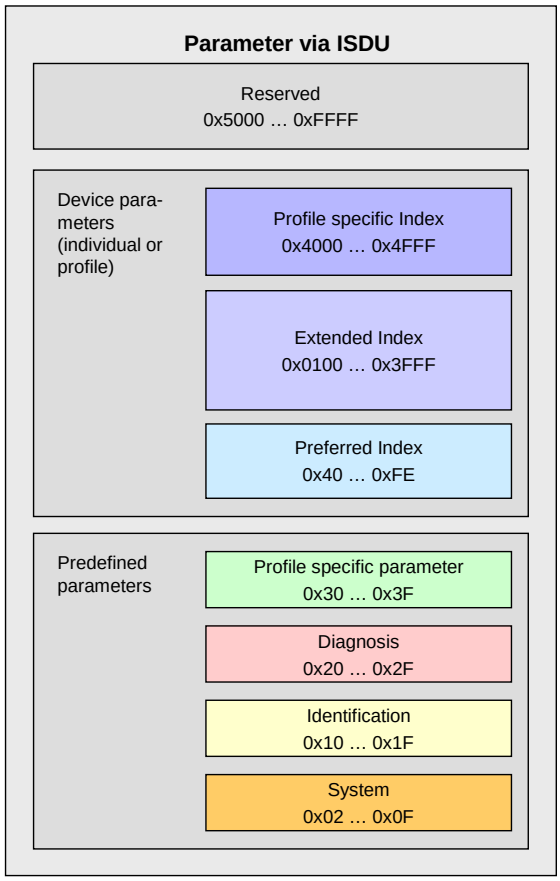


Figure B.6 – Index space for ISDU data objects

Clause B.2 contains definitions and requirements for the implementation of technology specific Device applications. Implementation rules for parameters and commands are specified in Table B.7.

Table B.7 – Implementation rules for parameters and commands

Rule number	Rule specification
1	All parameters of an Index shall be readable and/or writeable as an entire data object via Subindex 0
2	The technology specific device application shall resolve inconsistencies of dependent parameter sets during parameterization
3	The duration of an ISDU service request is limited (see Table 97). A master application can abort ISDU services after this timeout
4	Application commands (for example teach-in, reset to factory settings, etc.) are treated like parameters. The initiated execution of an application command is confirmed with a positive service response – Write.res(+). A negative service response – Write.res(-) – shall indicate that the execution of the application command failed. In both cases the timeout limit shall be considered (see Table 97)

Table B.8 specifies the assignment of data objects (parameters and commands) to the Index range of ISDUs. All indices above 2 are ISDU related.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table B.8 – Index assignment of data objects (Device parameter)

Index (dec)	Object name	Access	Length	Data type	M/O/C	Remark
0x0000 (0)	Direct Parameter Page 1	R		RecordT	M	Redirected to the page communication channel, see 10.7.5
0x0001 (1)	Direct Parameter Page 2	R/W		RecordT	M	Redirected to the page communication channel, see 10.7.5
0x0002 (2)	System-Command	W	1 octet	UIntegerT	M/O	Command Code Definition (See B.2.2)
0x0003 (3)	Data Storage Index	R/W	variable	RecordT	M	Set of data objects for storage (See B.2.3)
0x0004-0x000B (4-11)	Reserved					Reserved for exceptional operations
0x000C (12)	Device Access Locks	R/W	2 octets	RecordT	C	Standardized Device locking functions (See B.2.4)
0x000D (13)	Profile Characteristic	R	variable	ArrayT of UIntegerT16	O	Profile characteristic (see B.2.5)
0x000E (14)	PDInput Descriptor	R	variable	ArrayT of OctetStringT3	O	Reserved for Device profile (see B.2.6)
0x000F (15)	PDOutput Descriptor	R	variable	ArrayT of OctetStringT3	O	Reserved for Device profile (see B.2.7)
0x0010 (16)	Vendor Name	R	max. 64 octets	StringT	M	Informative (See B.2.8)
0x0011 (17)	Vendor Text	R	max. 64 octets	StringT	O	Additional vendor information (See B.2.9)
0x0012 (18)	Product Name	R	max. 64 octets	StringT	M	Detailed product or type name (See B.2.10)
0x0013 (19)	Product ID	R	max. 64 octets	StringT	O	Product or type identification (See B.2.11 for details)
0x0014 (20)	Product Text	R	max. 64 octets	StringT	O	Description of Device function or characteristic (See B.2.12)
0x0015 (21)	Serial-Number	R	max. 16 octets	StringT	O	Vendor specific serial number (See B.2.13)
0x0016 (22)	Hardware Revision	R	max. 64 octets	StringT	O	Vendor specific format (See B.2.14)
0x0017 (23)	Firmware Revision	R	max. 64 octets	StringT	O	Vendor specific format (See B.2.15)
0x0018 (24)	Application Specific Tag	R/W	Min. 16, max. 32 octets	StringT	O	Tag location or tag function defined by user (See B.2.16)
0x0019-0x001F (25-31)	Reserved					
0x0020 (32)	Error Count	R	2 octets	UIntegerT	O	Errors since power-on or reset (See B.2.17)
0x0021-0x0023 (33-35)	Reserved					
0x0024 (36)	Device Status	R	1 octet	UIntegerT	O	Contains current status of the Device (See B.2.18)
0x0025 (37)	Detailed Device Status	R	variable	ArrayT of OctetStringT3	O	See B.2.19

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Index (dec)	Object name	Access	Length	Data type	M/O/C	Remark
0x0026-0x0027 (38-39)	Reserved					
0x0028 (40)	Process-DataInput	R	PD length	Device specific	O	Read last valid Process Data from PDin channel (See B.2.19)
0x0029 (41)	Process-DataOutput	R	PD length	Device specific	O	Read last valid Process Data from PDout channel (See B.2.21)
0x002-0x002F (42-47)	Reserved					
0x0030 (48)	Offset Time	R/W	1 octet	RecordT	O	Synchronization of Device application timing to M-sequence timing (See B.2.22)
0x0031-0x003F (49-63)	Reserved for profiles					
0x0040-0x00FE (64-254)	Preferred Index					Device specific (8 bit)
0x00FF (255)	Reserved					
0x0100-0x3FFF (256-16 383)	Extended Index					Device specific (16 bit)
0x4000-0x4FFF (16 384-20 479)	Profile specific Index					Reserved for Device profile
0x5000-0xFFFF (20 480-65 535)	Reserved					
Key M = mandatory; O = optional; C = conditional						

B.2.2 SystemCommand

Devices with ISDU support shall use the ISDU Index 0x0002 to receive the SystemCommand. The commands shall be acknowledged. A positive acknowledge indicates the complete and correct finalization of the requested command. A negative acknowledge indicates the command cannot be realized or ended up with an error. A SystemCommand shall be executed within less than 5 s to fulfil the ISDU timing requirements (see Table 97).

Implementation of the SystemCommand feature is mandatory for Masters and optional for Devices. The coding of SystemCommand is specified in Table B.9.

Table B.9 – Coding of SystemCommand (ISDU)

Command (hex)	Command (dec)	Command name	M/O	Definition
0x00	0	Reserved		
0x01	1	ParamUploadStart	O	Start parameter upload
0x02	2	ParamUploadEnd	O	Stop parameter upload
0x03	3	ParamDownloadStart	O	Start parameter download
0x04	4	ParamDownloadEnd	O	Stop parameter download
0x05	5	ParamDownloadStore	O	Finalize parameterization and start Data Storage

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Command (hex)	Command (dec)	Command name	M/O	Definition
0x06	6	ParamBreak	O	Cancel all Param commands
0x07 to 0x3F	7 to 63	Reserved		
0x40 to 0x7F	64 to 127	Reserved for profiles		
0x80	128	Device reset	O	
0x81	129	Application reset	O	
0x82	130	Restore factory settings	O	
0x83 to 0x9F	131 to 159	Reserved		
0xA0 to 0xFF	160 to 255	Vendor specific		
NOTE See 10.3 **Key** M = mandatory; O = optional				

The SystemCommand 0x05 (ParamDownloadStore) shall be implemented according to 10.4.2, whenever the Device provides parameters to be stored via the Data Storage mechanism, i.e. parameter "Index_List" in Index 0x0003 is not empty (see Table B.10).

The implementation of the SystemCommands 0x01 to 0x06 required for block parameterization according to 10.3.5 is optional. However, all of these commands or none of them shall be implemented (for SystemCommand 0x05 the rule for Data Storage dominates).

See B.1.11 for SystemCommand options on the Direct Parameter page 1.

B.2.3 Data Storage Index

Table B.10 specifies the Data Storage Index assignments.

Table B.10 – Data Storage Index assignments

Index	Subindex	Access	Parameter Name	Coding	Data type
0x0003	01	R/W	DS_Command	0x00: Reserved 0x01: DS_UploadStart 0x02: DS_UploadEnd 0x03: DS_DownloadStart 0x04: DS_DownloadEnd 0x05: DS_Break 0x06 to 0xFF: Reserved	UintegerT8 (8 bit)
	02	R	State_Property	Bit 0: Reserved Bit 1 and 2: State of Data Storage 0b00: Inactive 0b01: Upload 0b10: Download 0b11: Data Storage locked Bit 3 to 6: Reserved Bit 7: DS_UPLOAD_FLAG "1": DS_UPLOAD_REQ pending "0": no DS_UPLOAD_REQ	UintegerT8 (8 bit)
	03	R	Data_Storage_ Size	Number of octets for storing all the necessary information for the Device replacement (see 10.4.5). Maximum size is 2 048 octets.	UintegerT16 (32 bit)
	04	R	Parameter_ Checksum	Parameter set revision indication: CRC signature or Revision Counter (see 10.4.8)	UintegerT32 (32 bit)
	05	R	Index_List	List of parameter indices to be saved (see Table B.11)	OctetStringT (variable)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

The parameter Data Storage Index 0x0003 contains all the information to be used for the Data Storage handling. This parameter is reserved for private exchanges between the Master and the Device; the Master shall block any access request from a gateway application to this Index (see Figure 4). The parameters within this Index 0x0003 are specified as follows.

DS_Command

This octet carries the Data Storage commands for the Device.

State_Property

This octet indicates the current status of the Data Storage mechanism. Bit 7 shall be stored in non-volatile memory. The Master checks this bit at start-up and performs a parameter upload if requested.

Data_Storage_Size

These four octets provide the requested memory size as number of octets for storing all the information required for the replacement of a Device including the structural information (Index, Subindex). Data type is UintegerT32 (32 bit). The maximum size is 2 048 octets. See Table F.1 for the elements to be taken into account in the size calculation.

Parameter_Checksum

This checksum is used to detect changes in the parameter set without reading all parameters. The value of the checksum is calculated according to the procedure in 10.4.8. The Device shall change the checksum whenever a parameter out of the parameter set has been altered. Different parameter sets shall hold different checksums. It is recommended that the Device stores this parameter locally in non-volatile memory.

Index_List

Table B.11 specifies the structure of the Index_List. Each Index_List can carry up to 70 entries (see Table 97).

Table B.11 – Structure of Index_List

Entry	Address	Definition	Data type
X1	Index	Index of first parameter to be saved	Unsigned16
	Subindex	Subindex of first parameter to be saved	Unsigned8
X2	Index	Index of next parameter to be saved	Unsigned16
	Subindex	Subindex of next parameter to be saved	Unsigned8
…..	…….	……	…….
Xn	Index	Index of last parameter to be saved	Unsigned16
	Subindex	Subindex of last parameter to be saved	Unsigned8
Xn+1	Index	Termination_Marker 0x0000: End of Index_List >0x0000: Next Index containing an Index_List	Unsigned16

Large sets of parameters can be handled via concatenated Index_Lists. The last two octets of the Index_List shall carry the Termination Marker. A value "0" indicates the end of the Index List. In case of concatenation the Termination Marker is set to the next Index containing an Index List. The structure of the following Index List is the same as specified in Table B.11. Thus, the concatenation of lists ends if a Termination Marker with the value "0" is found.

B.2.4 Device Access Locks

The parameter Device Access Locks allows control of the Device behaviour. Standardized Device functions can independently be configured via defined flags in this parameter. The Device Access Locks configuration can be changed by overwriting the parameter. The actual

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

configuration setting is available per read access to this parameter. The data type is RecordT of BooleanT. Access is only permitted via Subindex 0. This parameter is optional. If implemented it shall be non-volatile.

The following Device access lock categories are specified:

- Parameter write access (optional);
- Data Storage (mandatory if the Device supports Data Storage);
- local parameterization (optional);
- local user interface operation (optional).

Table B.12 lists the Device locking possibilities.

Table B.12 – Device locking possibilities

Bit	Category	Definition
0	Parameter (write) access (optional)	0: unlocked (default) 1: locked
1	Data Storage (mandatory if the Device supports Data Storage)	0: unlocked (default) 1: locked (see NOTE)
2	Local parameterization (optional)	0: unlocked (default) 1: locked
3	Local user interface (optional)	0: unlocked (default) 1: locked
4 – 15	Reserved	
NOTE The Master reads the parameter State_Property/State of Data Storage (see Table B.10) prior to any actions.		

Parameter (write) access

If this bit is set, write access to all Device parameters over the SDCI communication interface is inhibited for all read/write parameters of the Device except the parameter Device Access Locks. Read access is not affected. The Device shall respond with the negative service response – access denied – to a write access, if the parameter access is locked.

The parameter (write) access lock mechanism shall not block downloads of the Data Storage mechanism (between DS_DownloadStart and DS_DownloadEnd or DS_Break).

Data Storage

If this bit is set in the Device, the Data Storage mechanism is disabled (see 10.4.2 and 11.3.3). In this case, the Device shall respond to a write access (within its Data Storage Index) with a negative service response – access denied – (see B.2.3). Read access to its Data Storage Index is not affected.

This setting is also indicated in the State Property within Data Storage Index.

Local parameterization

If this bit is set, the parameterization via local control elements on the Device is inhibited.

Local user interface

If this bit is set, operation of the human machine interface on the Device is disabled.

B.2.5 Profile Characteristic

This parameter contains the list of ProfileIdentifiers (PID's) corresponding to the Device Profile implemented in the Device.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE Details are provided in [7].

B.2.6 PD Input Descriptor

This parameter contains the description of the data structure of the process input data for a profile Device.

NOTE Details are provided in [7].

B.2.7 PD Output Descriptor

This parameter contains the description of the data structure of the process output data for a profile Device.

NOTE Details are provided in [7].

B.2.8 Vendor Name

The parameter Vendor Name contains only one of the vendor names listed for the assigned VendorID. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is mandatory.

NOTE The list of vendor names associated with a given VendorID is maintained by the IO-Link consortium.

B.2.9 Vendor Text

The parameter Vendor Text contains additional information about the vendor. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is optional.

B.2.10 Product Name

The parameter Product Name contains the complete product name. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is mandatory.

NOTE The corresponding entry in the IODD Device variant list is expected to match this parameter.

B.2.11 Product ID

The parameter Product ID shall contain the vendor specific product or type identification of the Device. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is optional.

B.2.12 Product Text

The parameter Product Text shall contain additional product information for the Device, such as product category (for example Photoelectric Background Suppression, Ultrasonic Distance Sensor, Pressure Sensor, etc.). The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is optional.

B.2.13 SerialNumber

The parameter SerialNumber shall contain a unique vendor specific code for each individual Device. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 16. This parameter is optional.

B.2.14 Hardware Revision

The parameter Hardware Revision shall contain a vendor specific coding for the hardware revision of the Device. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is optional.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.15 Firmware Revision

The parameter Firmware Revision shall contain a vendor specific coding for the firmware revision of the Device. The parameter is a read-only data object. The data type is StringT with a maximum fixedLength of 64. This parameter is optional.

B.2.16 Application Specific Tag

The parameter Application Specific Tag shall be provided as read/write data object for the user application. It can serve as a "tag function" (role of the Device) or a "tag location" (location of the Device). The data type is StringT with a minimum fixedLength of 16 and a maximum fixedLength of 32. As default it is recommended to fill this parameter with "***". This parameter is optional.

NOTE In process automation usually this length is 32 octets.

B.2.17 Error Count

The parameter Error Count provides information on errors occurred in the Device application since power-on or reset. Usage of this parameter is vendor or Device specific. The data type is UIntegerT with a bitLength of 16. The parameter is a read-only data object. This parameter is optional.

B.2.18 Device Status

B.2.18.1 Overview

The parameter Device Status shall provide information about the Device condition (diagnosis) by the Device's technology. The data type is UIntegerT with a bitLength of 8. The parameter is a read-only data object. This parameter is optional.

The following Device conditions in Table B.13 are specified. They shall be generated by the Device applications. The parameter Device Status can be read by any PLC program or tools such as Asset Management (see Clause 11).

Table B.13 lists the different Device Status information. The criteria for these indications are specified in B.2.18.2 through B.2.18.5.

Table B.13 – Device status parameter

Value	Definition
0	Device is operating properly
1	Maintenance-Required (see B.2.18.2)
2	Out-of-Specification (see B.2.18.3)
3	Functional-Check (see B.2.18.4)
4	Failure (see B.2.18.5)
5 – 255	Reserved

B.2.18.2 Maintenance-required

Although the Process Data are valid, internal diagnostics indicate that the Device is close to loose its ability of correct functioning.

EXAMPLES Optical lenses getting dusty, build-up of deposits, lubricant level low.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.18.3 Out-of-Specification

Although the Process Data are valid, internal diagnostics indicate that the Device is operating outside its specified measuring range or environmental conditions.

EXAMPLES Power supply, auxiliary energy, temperature, pneumatic pressure, magnetic interference, vibrations, acceleration, interfering light, bubble formation in liquids.

B.2.18.4 Functional-Check

Process Data are temporarily invalid due to intended manipulations on the Device.

EXAMPLES Calibrations, teach-in, position adjustments, simulation.

B.2.18.5 Failure

Process Data invalid due to malfunction in the Device or its peripherals. The Device is unable to perform its intended function.

B.2.19 Detailed Device Status

The parameter Detailed Device Status shall provide information about currently pending Events in the Device. Events of TYPE "Error" or "Warning" and MODE "Event appears" (see A.6.4) shall be entered into the list of Detailed Device Status with EventQualifier and EventCode. Upon occurrence of an Event with MODE "Event disappears", the corresponding entry in Detailed Device Status shall be set to EventQualifier "0x00" and EventCode "0x0000". This way this parameter always provides the current diagnosis status of the Device. The parameter is a read-only data object. The data type is ArrayT with a maximum number of 64 array elements (Event entries). The number of array elements of this parameter is Device specific. Upon power-off or reset of the Device the contents of all array elements is set to initial settings – EventQualifier "0x00", EventCode "0x0000". This parameter is optional.

Table B.14 specifies the structure of the parameter Detailed Device Status.

Table B.14 – Detailed Device Status (Index 0x0025)

Subindex	Object name	Data Type	Comment
1	Error_Warning_1	3 octets	All octets 0x00: no Error/ Warning Octet 1: EventQualifier Octet 2,3: EventCode
2	Error_Warning_2	3 octets	
3	Error_Warning_3	3 octets	
4	Error_Warning_4	3 octets	
…			
n	Error_Warning_n	3 octets	

The designer may choose the implementation of a static list, i.e. one fix array position for each Event with a specific EventCode, or a dynamic list, i.e. each Event entry is stored into the next free array position. Subindex access is not permitted for a dynamic list.

B.2.20 ProcessDataInput

The parameter ProcessDataInput shall provide the last valid process input data from the Device application. The data type and structure is identical to the Process Data In transferred in the process communication channel. The parameter is a read-only data object. This parameter is optional.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.21 ProcessDataOutput

The parameter ProcessDataOutput shall provide the last valid process output data written to the Device application. The data type and structure is identical to the Process Data Out transferred in the process communication channel. The parameter is a read-only data object. This parameter is optional.

B.2.22 Offset Time

The parameter Offset Time allows a Device application to synchronize on M-sequence cycles of the data link layer via adjustable offset times. The data type is RecordT. Access is only possible via Subindex 0. The parameter is a read/write data object. This parameter is optional.

The structure of the parameter Offset Time is shown in Figure B.7:

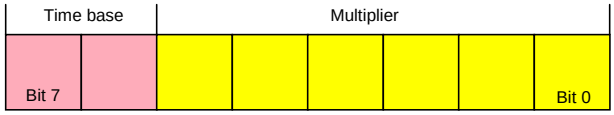


Figure B.7 – Structure of the Offset Time

Bits 0 to 5: Multiplier

These bits contain a 6-bit factor for the calculation of the Offset Time. Permissible values for the multiplier are 0 to 63.

Bits 6 to 7: Time Base

These bits contain the time base for the calculation of the Offset Time.

The permissible combinations for Time Base and Multiplier are listed in Table B.15 along with the resulting values for Offset Time. Setting both Multiplier and Time Base to zero deactivates synchronization with the help of an Offset Time. The value of Offset Time shall not exceed the MasterCycleTime (see B.1.3)

Table B.15 – Time base coding and values of Offset Time

Time base encoding	Time Base value	Calculation	Offset Time
00	0,01 ms	Multiplier × Time Base	0,01 ms to 0,63 ms
01	0,04 ms	0,64 ms + Multiplier × Time Base	0,64 ms to 3,16 ms
10	0,64 ms	3,20 ms + Multiplier × Time Base	3,20 ms to 43,52 ms
11	2,56 ms	44,16 ms + Multiplier × Time Base	44,16 ms to 126,08 ms

B.2.23 Profile Parameter (reserved)

Indices 0x0031 to 0x003F are reserved for Device profiles.

NOTE Details are provided in [7].

B.2.24 Preferred Index

Preferred Indices (0x0040 to 0x00FE) can be used for vendor specific Device functions. This range of indices is considered preferred due to lower protocol overhead within the ISDU and

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

thus higher data throughput for small data objects as compared to the Extended Index (see B.2.25).

B.2.25 Extended Index

Extended Indices (0x0100 to 0x3FFF) can be used for vendor specific Device functions.

B.2.26 Profile specific Index (reserved)

Indices 0x4000 to 0x4FFF are reserved for Device profiles.

NOTE Details are provided in [7].

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex C
(normative)

ErrorTypes (ISDU errors)

C.1 General

An ErrorType is used within negative service confirmations of ISDUs (see A.5.2 and Table A.13). It indicates the cause of a negative confirmation of a Read or Write service. The origin of the error may be located in the Master (local) or in the Device (remote).

The ErrorType consists of two octets, the main error cause and more specific information:

- ErrorCode (high order octet);
- AdditionalCode (low order octet).

The ErrorType represents information about the incident, the origin and the instance. The permissible ErrorTypes and the criteria for their deployment are listed in C.2 and C.3. All other ErrorType values are reserved and shall not be used.

C.2 Application related ErrorTypes

C.2.1 Overview

The permissible ErrorTypes resulting from the Device application are listed in Table C.1.

Table C.1 – ErrorTypes

Incident	Error Code	Additional Code	Name	Definition
Device application error – no details	0x80	0x00	APP_DEV	See C.2.2
Index not available	0x80	0x11	IDX_NOTAVAIL	See C.2.3
Subindex not available	0x80	0x12	SUBIDX_NOTAVAIL	See C.2.4
Service temporarily not available	0x80	0x20	SERV_NOTAVAIL	See C.2.5
Service temporarily not available – local control	0x80	0x21	SERV_NOTAVAIL_LOCCTRL	See C.2.6
Service temporarily not available – Device control	0x80	0x22	SERV_NOTAVAIL_DEVCTRL	See C.2.7
Access denied	0x80	0x23	IDX_NOT_WRITEABLE	See C.2.8
Parameter value out of range	0x80	0x30	PAR_VALOUTOFRNG	See C.2.9
Parameter value above limit	0x80	0x31	PAR_VALGTLIM	See C.2.10
Parameter value below limit	0x80	0x32	PAR_VALLTLIM	See C.2.11
Parameter length	0x80	0x33	VAL_LENOVRRUN	See C.2.12

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Incident	Error Code	Additional Code	Name	Definition
overrun				
Parameter length underrun	0x80	0x34	VAL_LENUNDRUN	See C.2.13
Function not available	0x80	0x35	FUNC_NOTAVAIL	See C.2.14
Function temporarily unavailable	0x80	0x36	FUNC_UNAVAILTEMP	See C.2.15
Invalid parameter set	0x80	0x40	PAR_SETINVALID	See C.2.16
Inconsistent parameter set	0x80	0x41	PAR_SETINCONSIST	See C.2.17
Application not ready	0x80	0x82	APP_DEVNOTRDY	See C.2.18
Vendor specific	0x81	0x00	UNSPECIFIC	See C.2.19
Vendor specific	0x81	0x01 to 0xFF	VENDOR_SPECIFIC	See C.2.19

C.2.2 Device application error – no details

This ErrorType shall be used if the requested service has been refused by the Device application and no detailed information of the incident is available.

C.2.3 Index not available

This ErrorType shall be used whenever a read or write access occurs to a not existing Index.

C.2.4 Subindex not available

This ErrorType shall be used whenever a read or write access occurs to a not existing Subindex.

C.2.5 Service temporarily not available

This ErrorType shall be used if a parameter is not accessible for a read or write service due to the current state of the Device application.

C.2.6 Service temporarily not available – local control

This ErrorType shall be used if a parameter is not accessible for a read or write service due to an ongoing local operation at the Device (for example operation or parameterization via an on-board Device control panel).

C.2.7 Service temporarily not available – device control

This ErrorType shall be used if a read or write service is not accessible due to a remote triggered state of the device application (for example parameterization during a remote triggered teach-in operation or calibration).

C.2.8 Access denied

This ErrorType shall be used if a write service tries to access a read-only parameter.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.2.9 Parameter value out of range

This ErrorType shall be used for a write service to a parameter outside its permitted range of values.

C.2.10 Parameter value above limit

This ErrorType shall be used for a write service to a parameter above its specified value range.

C.2.11 Parameter value below limit

This ErrorType shall be used for a write service to a parameter below its specified value range.

C.2.12 Parameter length overrun

This ErrorType shall be used when the content of a write service to a parameter is greater than the parameter specified length. This ErrorType shall also be used, if a data object is too large to be processed by the Device application (for example ISDU buffer restriction).

C.2.13 Parameter length underrun

This ErrorType shall be used when the content of a write service to a parameter is less than the parameter specified length (for example write access of an Unsigned16 value to an Unsigned32 parameter).

C.2.14 Function not available

This ErrorType shall be used for a write service with a command value not supported by the Device application (for example a SystemCommand with a value not implemented).

C.2.15 Function temporarily unavailable

This ErrorType shall be used for a write service with a command value calling a Device function not available due to the current state of the Device application (for example a SystemCommand).

C.2.16 Invalid parameter set

This ErrorType shall be used if values sent via single parameter transfer are not consistent with other actual parameter settings (for example overlapping set points for a binary data setting; see 10.3.4).

C.2.17 Inconsistent parameter set

This ErrorType shall be used at the termination of a block parameter transfer with ParamDownloadEnd or ParamDownloadStore if the plausibility check shows inconsistencies (see 10.3.5 and B.2.2).

C.2.18 Application not ready

This ErrorType shall be used if a read or write service is refused due to a temporarily unavailable application (for example peripheral controllers during startup).

C.2.19 Vendor specific

This ErrorType will be propagated directly to higher level processing elements as an error (no warning) by the Master.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.3 Derived ErrorTypes

C.3.1 Overview

Derived ErrorTypes are generated in the Master AL and are caused by internal incidents or those received from the Device. Table C.2 lists the specified Derived ErrorTypes.

Table C.2 – Derived ErrorTypes

Incident	Error Code	Additional Code	Name	Definition
Master – Communication error	0x10	0x00	COM_ERR	See C.3.2
Master – ISDU timeout	0x11	0x00	I-SERVICE_TIMEOUT	See C.3.3
Device Event – ISDU error (DL, Error, single shot, 0x5600)	0x11	0x00	I-SERVICE_TIMEOUT	See C.3.4
Device Event – ISDU illegal service primitive (AL, Error, single shot, 0x5800)	0x11	0x00	I-SERVICE_TIMEOUT	See C.3.5
Master – ISDU checksum error	0x56	0x00	M_ ISDU_CHECKSUM	See C.3.6
Master – ISDU illegal service primitive	0x57	0x00	M_ ISDU_ILLEGAL	See C.3.7
Device Event – ISDU buffer overflow (DL, Error, single shot, 0x5200)	0x80	0x33	VAL_LENOVRRUN	See C.3.8 and C.2.12 Events from legacy Devices shall be redirected in compatibility mode to this derived ErrorType

C.3.2 Master – Communication error

The Master generates a negative service response with this ErrorType if a communication error occurred during a read or write service, for example the SDCI connection is interrupted.

C.3.3 Master – ISDU timeout

The Master generates a negative service response with this ErrorType, if a Read or Write service is pending longer than the specified I-Service timeout (see Table 97) in the Master.

C.3.4 Device Event – ISDU error

If the Master received an Event with the EventQualifier (see A.6.4: DL, Error, Event single shot) and the EventCode 0x5600, a negative service response indicating a service timeout is generated and returned to the requester (see C.3.3).

C.3.5 Device Event – ISDU illegal service primitive

If the Master received an Event with the EventQualifier (see A.6.4: AL, Error, Event single shot) and the EventCode 0x5800, a negative service response indicating a service timeout is generated and returned to the requester (see C.3.3).

C.3.6 Master – ISDU checksum error

The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU checksum error.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.3.7 Master – ISDU illegal service primitive

The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU illegal service primitive.

C.3.8 Device Event – ISDU buffer overflow

If the Master received an Event with the EventQualifier (see A.6.4: DL, Error, Event single shot) and the EventCode 0x5200, a negative service response indicating a parameter length overrun is generated and returned to the requester (see C.2.12).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex D
(normative)

EventCodes (diagnosis information)

D.1 General

The concept of Events is described in 7.3.8.1 and the general structure and encoding of Events is specified in Clause A.6. Whenever the StatusCode indicates an Event in case of a Device or a Master incident, the associated EventCode shall be provided as diagnosis information. As specified in A.6, the Event entry contains an EventCode in addition to the EventQualifier. The EventCode identifies an actual incident. Permissible values for EventCode are listed in Table D.1; all other EventCode values are reserved and shall not be used.

D.2 EventCodes for Devices

Table D.1 lists the specified EventCode identifiers and their definitions. The EventCodes are created by the technology specific Device application (instance = APP).

Table D.1 – EventCodes

EventCodes	Definition and recommended maintenance action	Device Status Value (NOTE 1)	TYPE (NOTE 2)
0x0000	No malfunction	0	Notification
0x1000	General malfunction – unknown error	4	Error
0x1001 to 0x17FF	Reserved		
0x1800 to 0x18FF	Vendor specific		
0x1900 to 0x3FFF	Reserved		
0x4000	Temperature fault – Overload	4	Error
0x4001 to 0x420F	Reserved		
0x4210	Device temperature over-run – Clear source of heat	2	Warning
0x4211 to 0x421F	Reserved		
0x4220	Device temperature under-run – Insulate Device	2	Warning
0x4221 to 0x4FFF	Reserved		
0x5000	Device hardware fault – Device exchange	4	Error
0x5001 to 0x500F	Reserved		
0x5010	Component malfunction – Repair or exchange	4	Error
0x5011	Non volatile memory loss – Check batteries	4	Error
0x5012	Batteries low – Exchange batteries	2	Warning
0x5013 to 0x50FF	Reserved		
0x5100	General power supply fault – Check availability	4	Error
0x5101	Fuse blown/open – Exchange fuse	4	Error
0x5102 to 0x510F	Reserved		
0x5110	Primary supply voltage over-run – Check tolerance	2	Warning
0x5111	Primary supply voltage under-run – Check tolerance	2	Warning

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

EventCodes	Definition and recommended maintenance action	Device Status Value (NOTE 1)	TYPE (NOTE 2)
0x5112	Secondary supply voltage fault (Port Class B) – Check tolerance	2	Warning
0x5113 to 0x5FFF	Reserved		
0x6000	Device software fault – Check firmware revision	4	Error
0x6001 to 0x631F	Reserved		
0x6320	Parameter error – Check data sheet and values	4	Error
0x6321	Parameter missing – Check data sheet	4	Error
0x6322 to 0x634F	Reserved		
0x6350	Parameter changed – Check configuration	4	Error
0x6351 to 0x76FF	Reserved		
0x7700	Wire break of a subordinate device – Check installation	4	Error
0x7701 to 0x770F	Wire break of subordinate device 1 ...device 15 – Check installation	4	Error
0x7710	Short circuit – Check installation	4	Error
0x7711	Ground fault – Check installation	4	Error
0x7712 to 0x8BFF	Reserved		
0x8C00	Technology specific application fault – Reset Device	4	Error
0x8C01	Simulation active – Check operational mode	3	Warning
0x8C02 to 0x8C0F	Reserved		
0x8C10	Process variable range over-run – Process Data uncertain	2	Warning
0x8C11 to 0x8C1F	Reserved		
0x8C20	Measurement range over-run – Check application	4	Error
0x8C21 to 0x8C2F	Reserved		
0x8C30	Process variable range under-run – Process Data uncertain	2	Warning
0x8C31 to 0x8C3F	Reserved		
0x8C40	Maintenance required – Cleaning	1	Notification
0x8C41	Maintenance required – Refill	1	Notification
0x8C42	Maintenance required – Exchange wear and tear parts	1	Notification
0x8C43 to 0x8C9F	Reserved		
0x8CA0 to 0x8DFF	Vendor specific		
0x8E00 to 0xAFFF	Reserved		
0xB000 to 0xBFFF	Reserved for profiles		
0xC000 to 0xFEFF	Reserved		
0xFF00 to 0xFFFF	SDCI specific EventCodes (see Table D.2)		

NOTE 1 See B.2.18.
NOTE 2 See Table A.19.

Table D.2 lists basic SDCI Events related to system management, Device or Master application, and specifies how they are encoded. Other types of Events may be reported but are not specified in this standard. Processing of these Events by the Master is vendor specific.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table D.2 – Basic SDCI EventCodes

Incident [a]	Origin	Instance	Name	EventCode	Action	Remark
System management						
Mode indication	LOCAL	DL	NEW_SLAVE	0xFF21	PD stop	See Clause 11
Device communi-cation lost	LOCAL	APP	DEV_COM_ LOST	0xFF22	-	See Clause 11
Data Storage identification mismatch	LOCAL	APP	DS_IDENT_ MISMATCH	0xFF23	-	See Clause 11
Data Storage buffer overflow	LOCAL	APP	DS_BUFFER _OVERFLOW	0xFF24	-	See Clause 11
Data Storage parameter access denied	LOCAL	APP	DS_ACCESS _DENIED	0xFF25	-	See Clause 11
Unspecified						
Incorrect Event signalling	LOCAL	DL	EVENT	0xFF31	Event.ind	See Clause 11
Device specific application						
Data Storage upload request	REMOTE	APP	DS_UPLOAD _REQ	0xFF91	Event.ind	
Reserved	REMOTE	APP		0xFF98	Event.ind	Shall not be used

[a] All Events are of StatusCode type 2 (with details), EventQualifier type "Notification", EventQualifier Mode "Single-shot".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex E
(normative)

Data types

E.1 General

This annex specifies basic and composite data types. Examples demonstrate the structures and the transmission aspects of data types for singular use or in a packed manner.

NOTE More examples are available in [6].

E.2 Basic data types

E.2.1 General

The coding of basic data types is shown only for singular use, which is characterized by:

- Process Data consisting of one basic data type;
- Parameter consisting of one basic data type;
- Subindex (>0) access on individual data items of parameters of composite data types (arrays, records).

E.2.2 BooleanT

A BooleanT is representing a data type that can have only two different values i.e. TRUE and FALSE. The data type is specified in Table E.1. For singular use the coding is shown in Table E.2. A sender shall always use 0xFF for 'TRUE' or 0x00 for 'FALSE'. A receiver can interpret the range from 0x01 through 0xFF for 'TRUE' and shall interpret 0x00 for 'FALSE' to simplify implementations. The packed form is demonstrated in Table E.22 and Figure E.8.

Table E.1 – BooleanT

Data type name	Value range	Resolution	Length
BooleanT	TRUE / FALSE	-	1 bit or 1 octet

Table E.2 – BooleanT coding

Bit	7	6	5	4	3	2	1	0	Values
TRUE	1	1	1	1	1	1	1	1	0xFF
FALSE	0	0	0	0	0	0	0	0	0x00

E.2.3 UIntegerT

A UIntegerT is representing an unsigned number depicted by 2 up to 64 bits ("enumerated"). The number is accommodated and right-aligned within the following permitted octet containers: 1, 2, 4, or 8. High order padding bits are filled with "0". Coding examples are shown in Figure E.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

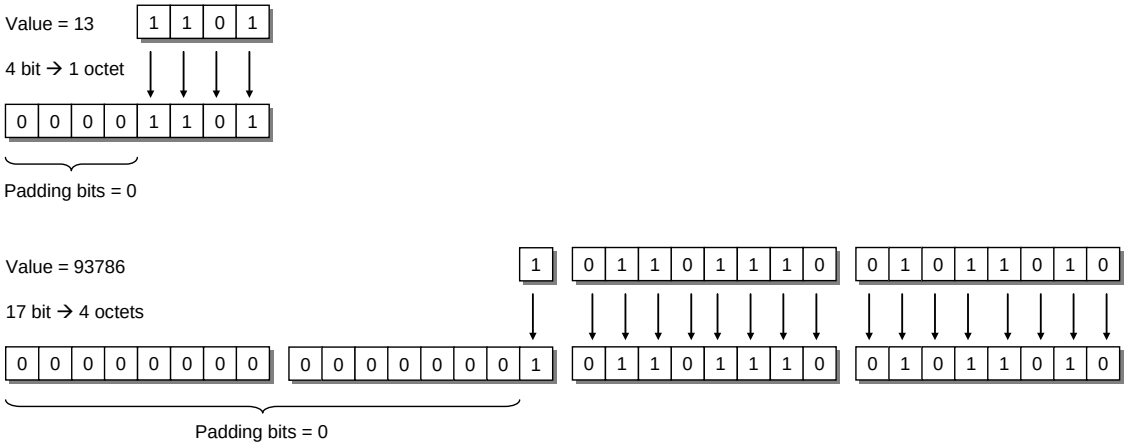


Figure E.1 – Coding examples of UIntegerT

The data type UIntegerT is specified in Table E.3 for singular use.

Table E.3 – UIntegerT

Data type name	Value range	Resolution	Length
UIntegerT	0 … $2^{bitlength} - 1$	1	1 octet, or 2 octets, or 4 octets, or 8 octets
NOTE 1 High order padding bits are filled with "0". NOTE 2 Most significant octet (MSO) sent first.			

E.2.4 IntegerT

An IntegerT is representing a signed number depicted by 2 up to 64 bits. The number is accommodated within the following permitted octet containers: 1, 2, 4, or 8 and right-aligned and extended correctly signed to the chosen number of bits. The data type is specified in Table E.4 for singular use. SN represents the sign with "0" for all positive numbers and zero, and "1" for all negative numbers. Padding bits are filled with the content of the sign bit (SN).

Table E.4 – IntegerT

Data type name	Value range	Resolution	Length
IntegerT	$-2^{bitlength\ -1}$ … $2^{bitlength\ -1} - 1$	1	1 octet, or 2 octets, or 4 octets, or 8 octets
NOTE 1 High order padding bits are filled with the value of the sign bit (SN). NOTE 2 Most significant octet (MSO) sent first (lowest respective octet number in Table E.5).			

The 4 coding possibilities in containers are listed in Table E.5 through Table E.8.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table E.5 – IntegerT coding (8 octets)

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	8 octets
Octet 2	2^{55}	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Octet 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Octet 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Table E.6 – IntegerT coding (4 octets)

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	4 octets
Octet 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Table E.7 – IntegerT coding (2 octets)

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2 octets
Octet 2	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Table E.8 – IntegerT coding (1 octet)

Bit	7	6	5	4	3	2	1	0	Container
Octet 1	SN	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	1 octet

Coding examples within containers are shown in Figure E.2

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

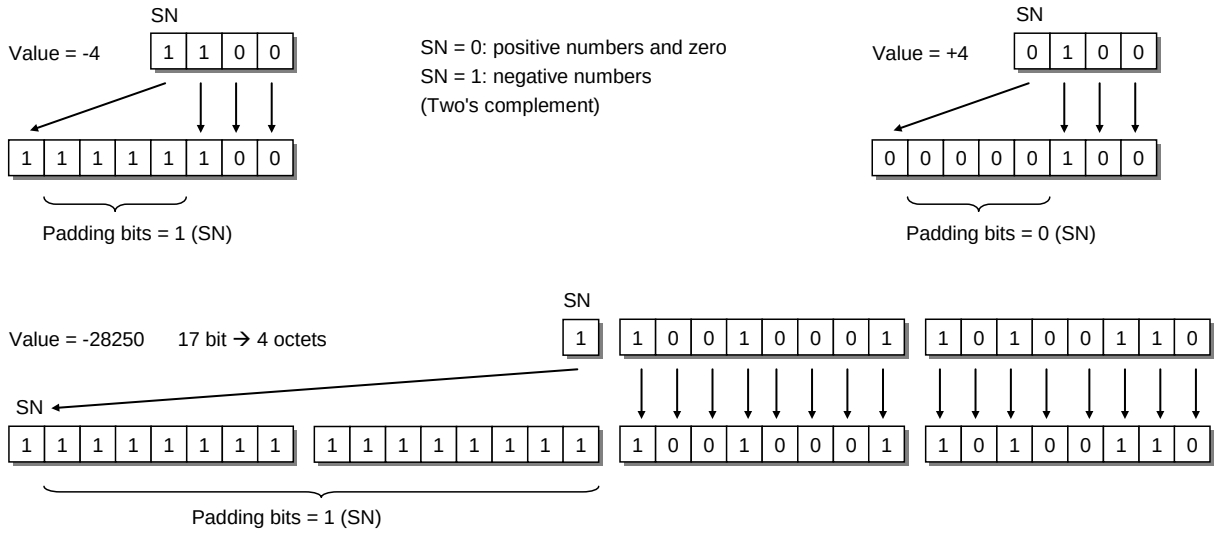


Figure E.2 – Coding examples of IntegerT

E.2.5 Float32T

A Float32T is representing a number specified by IEEE Std 754-2008 as single precision (32 bit). Table E.9 gives the definition and Table E.10 the coding. SN represents the sign with "0" for all positive numbers and zero, and "1" for all negative numbers.

Table E.9 – Float32T

Data type name	Value range	Resolution	Length
Float32T	See IEEE Std 754-2008	See IEEE Std 754-2008	4 octets

Table E.10 – Coding of Float32T

Bits	7	6	5	4	3	2	1	0
Octet 1	SN	Exponent (E)						
	2^0	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 2	(E)	Fraction (F)						
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Octet 3	Fraction (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Octet 4	Fraction (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

In order to realize negative exponent values a special exponent encoding mechanism is set in place as follows:

The Float32T exponent (E) is encoded using an offset binary representation, with the zero offset being 127; also known as exponent bias in IEEE Std 754-2008.

E_{min} = 0x01 - 0x7F = -126

E_{max} = 0xFE - 0x7F = 127

Exponent bias = 0x7F = 127

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Thus, as defined by the offset binary representation, in order to get the true exponent the offset of 127 shall be subtracted from the stored exponent.

E.2.6 StringT

A StringT is representing an ordered sequence of symbols (characters) with a variable or fixed length of octets (maximum of 232 octets) coded in US-ASCII (7 bit) or UTF-8. UTF-8 uses one octet for all ASCII characters and up to 4 octets for other characters. 0x00 is not permitted as a character. Table E.11 gives the definition.

Table E.11 – StringT

Data type name	Encoding	Standards	Length
StringT	US-ASCII	see ISO/IEC 646	Any length of character string with a maximum of 232 octets
	UTF-8	see ISO/IEC 10646	
NOTE The length may be obtained from a Device's IODD via the attribute 'fixedLength'.			

An instance of StringT can be shorter than defined by the IODD attribute 'fixedLength'. 0x00 shall be used for the padding of unused octets. Character strings can be transmitted in their actual length in case of singular access (see Figure E.3). Optimization for transmission is possible by omitting the padding octets if the IODD attribute 'fixedLengthRestriction' is not set. The receiver can deduce the original length from the length of the ISDU or by searching the first NULL (0x00) character (See A.5.2 and A.5.3).

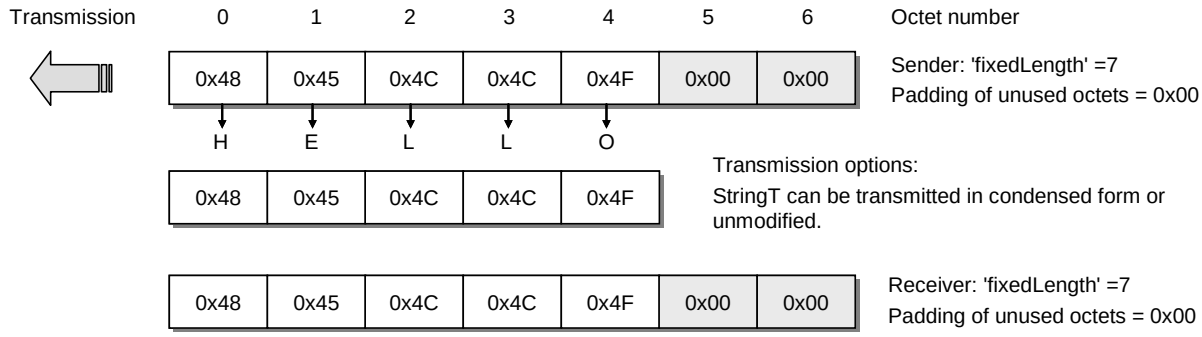


Figure E.3 – Singular access of StringT

E.2.7 OctetStringT

An OctetStringT is representing an ordered sequence of octets with a fixed length (maximum of 232 octets). Table E.12 gives the definition and Figure E.4 a coding example for a fixed length of 7.

Table E.12 – OctetStringT

Data type name	Value range	Standards	Length
OctetStringT	0x00 ... 0xFF per octet	-	Fixed length with a maximum of 232 octets
NOTE The length may be obtained from a Device's IODD via the attribute 'fixedLength'.			

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

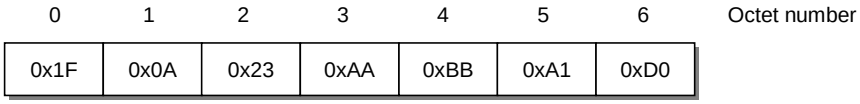


Figure E.4 – Coding example of OctetStringT

E.2.8 TimeT

A TimeT is based on the RFC 5905 standard and composed of two unsigned values that express the network time related to a particular date. Its semantic has changed from RFC 5905 according to Figure E.5. Table E.13 gives the definition and Table E.14 the coding of TimeT.

The first element is a 32-bit unsigned integer data type that provides the network time in seconds since 1900-01-01 0.00,00(UTC) or since 2036-02-07 6.28,16(UTC) for time values less than 0x9DFF4400, which represents the 1984-01-01 0:00,00(UTC). The second element is a 32-bit unsigned integer data type that provides the fractional portion of seconds in $1/2^{32}$ s. Rollovers after 136 years are not automatically detectable and shall be maintained by the application.

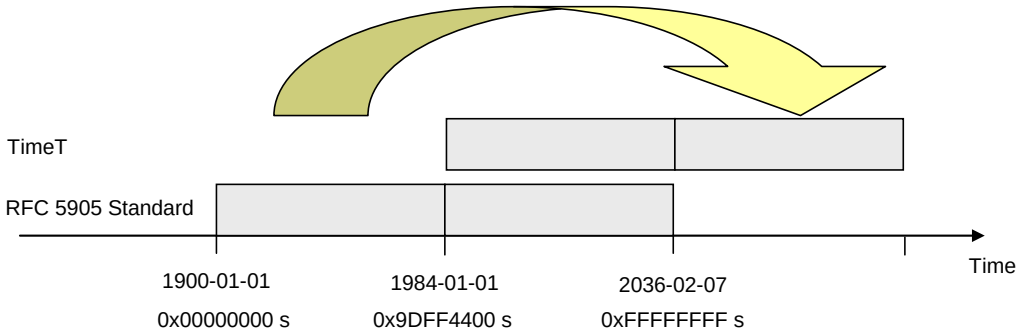


Figure E.5 – Definition of TimeT

Table E.13 – TimeT

Data type name	Value range	Resolution	Length
TimeT	Octet 1 to 4 (see Table E.14): $0 \leq i \leq (2^{32}-1)$	s (Seconds)	8 Octets (32 bit unsigned integer + 32 bit unsigned integer)
	Octet 5 to 8 (see Table E.14): $0 \leq i \leq (2^{32}-1)$	$(1/2^{32})$ s	
NOTE 32 bit unsigned integer are normal computer science data types.			

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table E.14 – Coding of TimeT

Bit	7	6	5	4	3	2	1	0	Definitions
Octet 1	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Seconds since 1900-01-01 0.00,00 or since 2036-02-07 6.28,16 when time value less than 0x9DFF4400.00000000
Octet 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Fractional part of seconds. One unit is $1/(2^{32})$ s
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = Most significant bit LSB = Least significant bit

E.2.9 TimeSpanT

A TimeSpanT is a 64-bit integer value i.e. a two's complement binary number with a length of eight octets, providing the network time difference in fractional portion of seconds in $1/2^{32}$ seconds. Table E.15 gives the definition and Table E.16 the coding of TimeSpanT.

Table E.15 – TimeSpanT

Data type name	Value range	Resolution	Length
TimeSpanT	Octet 1 to 8 (see Table E.16): $-2^{63} \leq i \leq (2^{63}-1)$	$(1/2^{32})$ s	8 octets (64 bit integer)
NOTE 64 bit integer is a normal computer science data type.			

Table E.16 – Coding of TimeSpanT

Bit	7	6	5	4	3	2	1	0	Definitions
Octet 1	2^{63}	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	Fractional part of seconds as 64 bit integer. One unit is $1/(2^{32})$ s.
Octet 2	2^{55}	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Octet 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Octet 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = Most significant bit LSB = Least significant bit

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

E.3 Composite data types

E.3.1 General

Composite data types are combinations of basic data types only. A composite data type consists of several basic data types packed within a sequence of octets. Unused bit space shall be padded with "0".

E.3.2 ArrayT

An ArrayT addressed by an Index is a data structure with data items of the same data type. The individual data items are addressable by the Subindex. Subindex 0 addresses the whole array within the Index space. The structuring rules for arrays are given in Table E.17.

Table E.17 – Structuring rules for ArrayT

Rule number	Rule specification
1	The Subindex data items are packed in a row without gaps describing an octet sequence
2	The highest Subindex data item *n* starts right-aligned within the octet sequence
3	UIntegerT and IntegerT with a length of ≥ 58 bit and < 64 bit are not permitted

Table E.18 and Figure E.6 give an example for the access of an array. Its content is a set of parameters of the same basic data type.

Table E.18 – Example for the access of an ArrayT

Index	Subindex	Offset	Data items	Data Type
66	1	12	0x2	IntegerT, 'bitLength' = 3
	2	9	0x6	
	3	6	0x4	
	4	3	0x7	
	5	0	0x5	

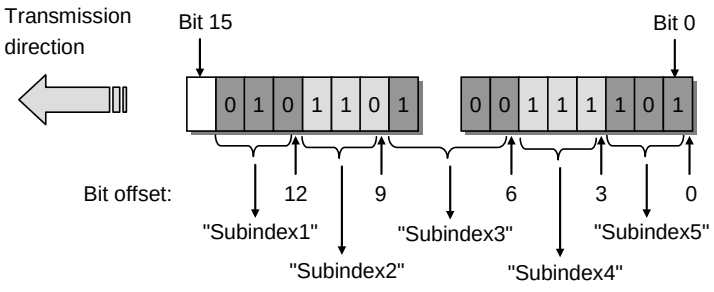


Figure E.6 – Example of an ArrayT data structure

E.3.3 RecordT

A record addressed by an Index is a data structure with data items of different data types. The Subindex allows addressing individual data items within the record on certain bit positions.

NOTE Bit positions within a RecordT may be obtained from the IODD of the particular Device.

The structuring rules for records are given in Table E.19.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Table E.19 – Structuring rules for RecordT

Rule number	Rule specification
1	The Subindices within the IODD shall be listed in ascending order from 1 to *n* describing an octet sequence. Gaps within the list of Subindices are allowed
2	Bit offsets shall always be indicated within this octet sequence (may show no strict order in the IODD)
3	The bit offset starts with the last octet within the sequence; this octet starts with offset 0 for the least significant bit and offset 7 for the most significant bit
4	The following data types shall always be aligned on octet boundaries: Float32T, StringT, OctetStringT, TimeT, and TimeSpanT
5	UIntegerT and IntegerT with a length of ≥ 58 bit shall always be aligned on one side of an octet boundary
6	It is highly recommended for UIntegerT and IntegerT with a length of ≥ 8 bit to align always on one side of an octet boundary
7	It is highly recommended for UIntegerT and IntegerT with a length of < 8 bit not to cross octet boundaries
8	A bit position shall not be used by more than one record item

Table E.20 gives an example 1 for the access of a RecordT. It consists of varied parameters named "Status", "Text", and "Value".

Table E.20 – Example 1 for the access of a RecordT

Index	Subindex	Offset	Data items							Data Type	Name
47	1	88	0x23	0x45						UIntegerT, 'bitLength' = 16	Status
	2	32	H	E	L	L	O	0x00	0x00	StringT, 'fixedLength' = 7	Text
	3	0	0x56	0x12	0x22	0x34				UIntegerT, 'bitLength' = 32	Value
NOTE 'bitLength' and 'fixedLength' are defined in the IODD of the particular Device.											

Table E.21 gives an example 2 for the access of a RecordT. It consists of varied parameters named "Level", "Min", and "Max". Figure E.7 shows the corresponding data structure.

Table E.21 – Example 2 for the access of a RecordT

Index	Subindex	Offset	Data items			Data Type	Name
46	1	2	0x32	0xF1		UIntegerT, 'bitLength' = 14	Level
	2	1	FALSE			BooleanT	Min
	3	0	TRUE			BooleanT	Max
NOTE 'bitLength' is defined in the IODD of the particular Device.							

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

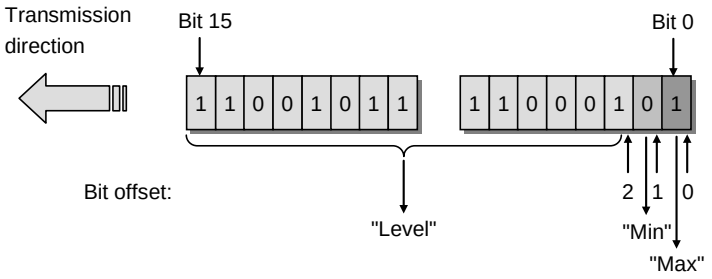


Figure E.7 – Example 2 of a RecordT structure

Table E.22 gives an example 3 for the access of a RecordT. It consists of varied parameters named "Control" through "Enable". Figure E.8 demonstrates the corresponding RecordT structure of example 3 with the bit offsets.

Table E.22 – Example 3 for the access of a RecordT

Index	Subindex	Offset	Data items		Data Type	Name
45	1	32	TRUE		BooleanT	NewBit
	2	33	FALSE		BooleanT	DR4
	3	34	FALSE		BooleanT	CR3
	4	35	TRUE		BooleanT	CR2
	5	38	TRUE		BooleanT	Control
	6	16	0xF8	0x23	OctetStringT, 'fixedLength' = 2	Setpoint
	7	8	0x41		StringT, 'fixedLength' = 1	Unit
	8	0	0xC3		OctetStringT, 'fixedLength' = 1	Enable
NOTE 'fixedLength' is defined in the IODD of the particular Device						

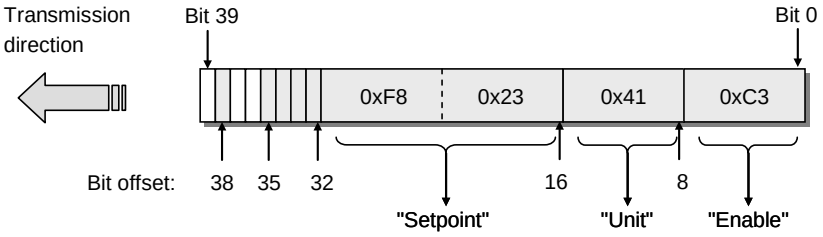


Figure E.8 – Example 3 of a RecordT structure

Figure E.9 shows a selective write request of a variable within the RecordT of example 3 and a write request of the complete RecordT (see A.5.7).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

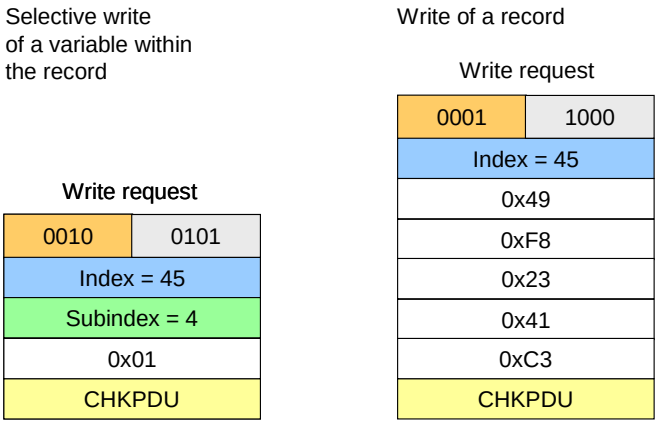


Figure E.9 – Write requests for example 3

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex F
(normative)

Structure of the Data Storage data object

Table F.1 gives the structure of a Data Storage (DS) data object within the Master (see 11.3.2).

Table F.1 – Structure of the stored DS data object

Part	Parameter name	Definition	Data type
Object 1	ISDU_Index	ISDU Index (0 to 65 535)	Unsigned16
	ISDU_Subindex	ISDU Index (0 to 255)	Unsigned8
	ISDU_Length	Length of the subsequent record	Unsigned8
	ISDU_Data	Record of length ISDU_Length	Record
Object 2	ISDU_Index	ISDU Index (0 to 65 535)	Unsigned16
	ISDU_Subindex	ISDU Index (0 to 255)	Unsigned8
	ISDU_Length	Length of the subsequent record	Unsigned8
	ISDU_Data	Record of length ISDU_Length	Record

Object *n*	ISDU_Index	ISDU Index (0 to 65 535)	Unsigned16
	ISDU_Subindex	ISDU Index (0 to 255)	Unsigned8
	ISDU_Length	Length of the subsequent record	Unsigned8
	ISDU _Data	Record of length ISDU_Length	Record

The Device shall calculate the required memory size by summarizing the objects 1 to *n* (see Table B.10, Subindex 3).

The Master shall store locally in non-volatile memory the header information specified in Table F.2. See Table B.10.

Table F.2 – Associated header information for stored DS data objects

Part	Parameter name	Definition	Data type
Header	Parameter Checksum	32 bit CRC signature or revision counter (see 10.4.8)	Unsigned32
	VendorID	See B.1.8	Unsigned16
	DeviceID	See B.1.9	Unsigned32
	FunctionID	See B.1.10	Unsigned16

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex G
(normative)

Master and Device conformity

G.1 Electromagnetic compatibility requirements (EMC)

G.1.1 General

The EMC requirements of this annex are only relevant for the SDCI interface part of a particular Master or Device. The technology functions of a Device and its relevant EMC requirements are not in the scope of this standard. For this purpose the Device specific product standards shall apply. For Master usually the EMC requirements for peripherals are specified in IEC 61131-2 or IEC 61000-6-2.

To ensure proper operating conditions of the SDCI interface, the test configurations specified in G.1.6 (Master) or G.1.7 (Device) shall be maintained during all the EMC tests. The tests required in the product standard of equipment under test (EUT) can alternatively be performed in SIO mode.

G.1.2 Operating conditions

It is highly recommended to evaluate the SDCI during the startup phase with the cycle times given in Table G.1. In most cases, this leads to the minimal time requirements for the performance of these tests. Alternatively, the SDCI may be evaluated during normal operation of the Device, provided that the required number of M-sequences specified in Table G.1 took place during each test.

G.1.3 Performance criteria

a) Performance criterion A

The SDCI operating at an average cycle time as specified in Table G.1 shall not show more than six detected M-sequence errors within the number of M-sequences given in Table G.1. No interruption of communication is permitted.

Table G.1 – EMC test conditions for SDCI

Transmission rate	Master		Device		Maximum of M-sequence errors
	t_{CYC}	Number of M-sequences of TYPE_2_5 (read) (6 octets)	t_{CYC}	Number of M-sequences of TYPE_0 (read) (4 octets)	
4,8 kbit/s	18,0 ms	300 (6 000)	100 T_{BIT} (20,84 ms)	350 (7 000)	6
38,4 kbit/s	2,3 ms	450 (9 000)	100 T_{BIT} (2,61 ms)	500 (10 000)	6
230,4 kbit/s	0,4 ms	700 (14 000)	100 T_{BIT} (0,44 ms)	800 (16 000)	6
NOTE The numbers of M-sequences are calculated according to the algorithm in H.2 and rounded up. The larger number of M-sequences (in brackets) are required if a certain test (for example fast transients/burst) applies interferences only with a burst/cycle ratio (see Table G.2)					

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

b) Performance Criterion B

The error rate of criterion A shall also be satisfied after but not during the test. No change of actual operating state (e.g. permanent loss of communication) or stored data is allowed.

G.1.4 Required immunity tests

Table G.2 specifies the EMC tests to be performed.

Table G.2 – EMC test levels

Phenomena	Test Level	Performance Criterion	Constraints
Electrostatic discharges (ESD) IEC 61000-4-2	Air discharge: ± 8 kV Contact discharge: ± 4 kV	B	See G.1.4, a)
Radio-frequency electromagnetic field. Amplitude modulated IEC 61000-4-3	80 MHz – 1 000 MHz 10 V/m 1 400 MHz – 2 000 MHz 3 V/m 2 000 MHz – 2 700 MHz 1 V/m	A	See G.1.4, a) and G.1.4, b)
Fast transients (Burst) IEC 61000-4-4	± 1 kV	A	5 kHz only. The number of M-sequences in Table G.1 shall be increased by a factor of 20 due to the burst/cycle ratio 15 ms/300 ms. See G.1.4, c)
	± 2 kV	B	
Surge IEC 61000-4-5	Not required for an SDCI link (SDCI link is limited to 20 m)		-
Radio-frequency common mode IEC 61000-4-6	0,15 MHz – 80 MHz 10 VEMF	A	See G.1.4, b) and G.1.4, d)
Voltage dips and interruptions IEC 61000-4-11	Not required for an SDCI link		

The following requirements also apply as specified in Table G.2.

a) As this phenomenon influences the entire device under test, an existing device specific product standard shall take precedence over the test levels specified here.

b) The test shall be performed with a step size of 1 % and a dwell of 1 s. If a single M-sequence error occurs at a certain frequency, that frequency shall be tested until the number of M-sequences according to Table G.1 has been transmitted or until 6 M-sequence errors have occurred.

c) Depending on the transmission rate the test time varies. The test time shall be at least one minute (with the transmitted M-sequences and the permitted errors increased accordingly).

d) This phenomenon is expected to influence most probably the EUTs internal analog signal processing and only with a very small probability the functionality of the SDCI communication. Therefore an existing device specific product standard shall take precedence over the test levels specified here.

G.1.5 Required emission tests

The definition of emission limits is not in the scope of this standard. The requirements of the Device specific product family or generic standards apply, usually for general industrial environments the IEC 61000-6-4.

All emission tests shall be performed at the fastest possible communication rate with the fastest cycle time.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

G.1.6 Test configurations for Master

G.1.6.1 General rules

The following rules apply for the test of Masters.

- In the following test setup diagrams only the SDCI and the power supply cables are shown. All other cables shall be treated as required by the relevant product standard.
- Grounding of the Master and the Devices shall be according to the relevant product standard or manual.
- Where not otherwise stated, the SDCI cable shall have an overall length of 20 m. Excess length laid as an inductive coil with a diameter of 0,3 m, where applicable mounted 0,1 m above reference ground.
- Where applicable, the auxiliary Devices shall be placed 10 cm above RefGND.
- A typical test configuration consists of the Master and two Devices, except for the RF common mode test, where only one Device shall be used.
- Each port shall fulfill the EMC requirements.

G.1.6.2 Electrostatic discharges

Figure G.1 shows the test setup for electrostatic discharge according to IEC 61000-4-2.

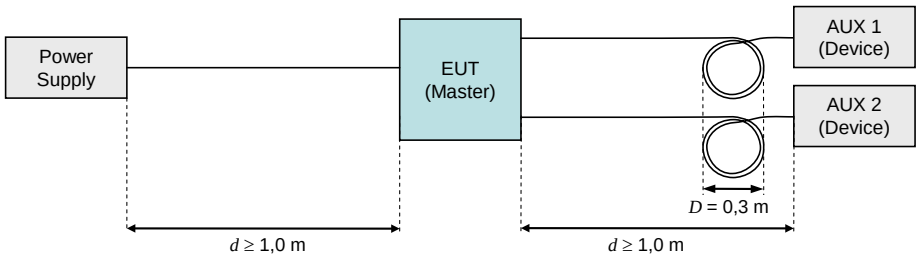


Figure G.1 – Test setup for electrostatic discharge (Master)

G.1.6.3 Radio-frequency electromagnetic field

Figure G.2 shows the test setup for radio-frequency electromagnetic field according to IEC 61000-4-3.

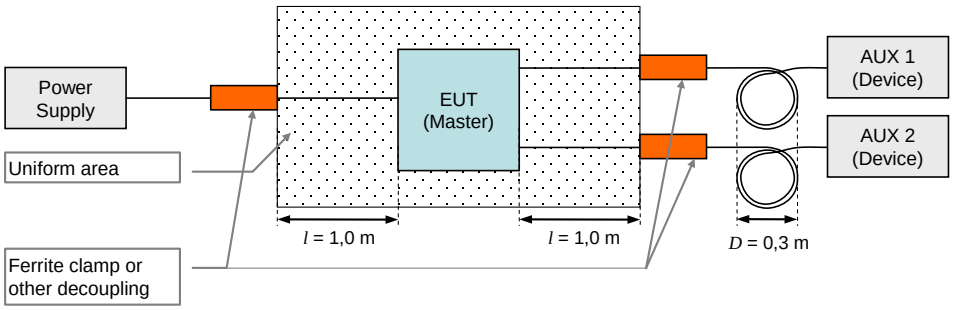
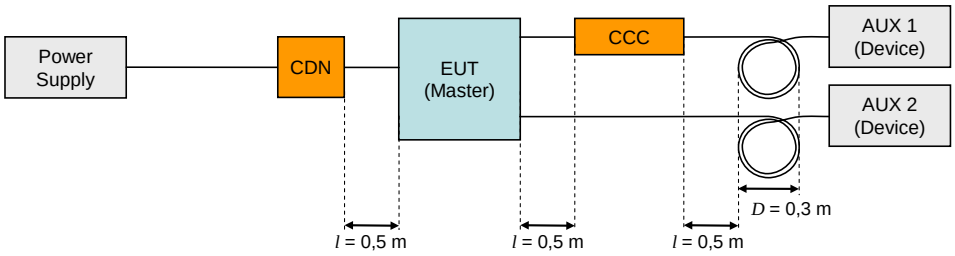


Figure G.2 – Test setup for RF electromagnetic field (Master)

G.1.6.4 Fast transients (burst)

Figure G.3 shows the test setup for fast transients according to IEC 61000-4-4. No coupling into SDCI line to AUX 2 is required.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



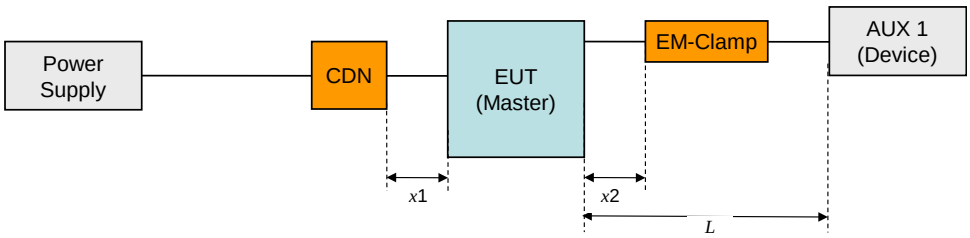
Key

CDN:Coupling/Decoupling Network
CCC: Capacitive coupling clamp

Figure G.3 – Test setup for fast transients (Master)

G.1.6.5 Radio-frequency common mode

Figure G.4 shows the test setup for radio-frequency common mode according to IEC 61000-4-6.



Key

0,1 m ≤ × 1 ≤ 0,3 m
0,1 m ≤ × 2 ≤ 0,3 m
L = 1,0 m ± 0,05 m

Figure G.4 – Test setup for RF common mode (Master)

G.1.7 Test configurations for Devices

G.1.7.1 General rules

For the test of Devices the following rules apply.

- In the following test setup diagrams only the SDCI and the power supply cables are shown. All other cables shall be treated as required by the relevant product standard.
- Grounding of the Master and the Devices according to the relevant product standard or user manual.
- Where not otherwise stated, the SDCI cable shall have an overall length of 20 m. Excess length laid as an inductive coil with a diameter of 0,3 m, where applicable mounted 0,1 m above RefGND.
- Where applicable, the auxiliary Devices shall be placed 10 cm above RefGND.
- Test with Device AUX 2 is optional.

G.1.7.2 Electrostatic discharges

Figure G.5 shows the test setup for electrostatic discharge according to IEC 61000-4-2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

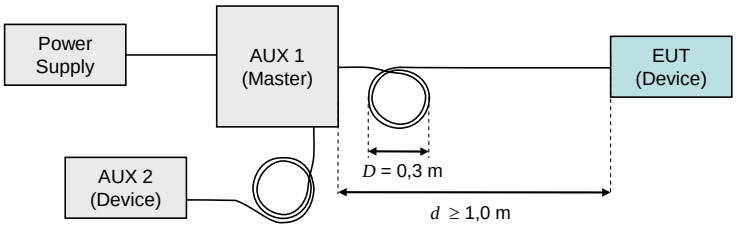


Figure G.5 – Test setup for electrostatic discharges (Device)

G.1.7.3 Radio-frequency electromagnetic field

Figure G.6 shows the test setup for radio-frequency electromagnetic field according to IEC 61000-4-3.

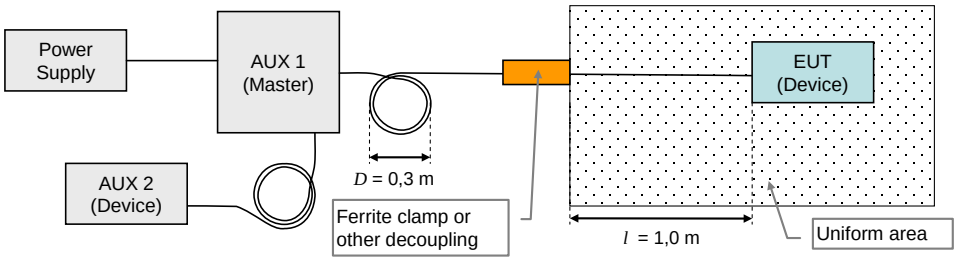
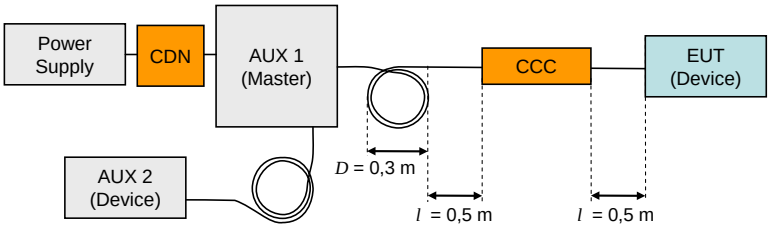


Figure G.6 – Test setup for RF electromagnetic field (Device)

G.1.7.4 Fast transients (burst)

Figure G.7 shows the test setup for fast transients according to IEC 61000-4-4.



Key

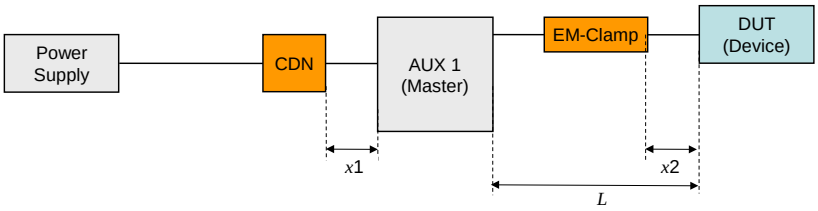
CDN: Coupling/Decoupling Network, here only used for decoupling
CCC: Capacitive coupling clamp

Figure G.7 – Test setup for fast transients (Device)

G.1.7.5 Radio-frequency common mode

Figure G.8 shows the test setup for radio-frequency common mode according to IEC 61000-4-6.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Key

0,1 m ≤ × 1 ≤ 0,3 m
0,1 m ≤ × 2 ≤ 0,3 m
L = 1,0 m ± 0,05 m

Figure G.8 – Test setup for RF common mode (Device)

G.2 Test strategies for conformity

G.2.1 Test of a Device

The Master AUX 1 (see Figure G.5) shall continuously send an M-sequence TYPE_0 (read Direct Parameter page 2) message at the cycle time specified in Table G.1 and count the missing and the erroneous Device responses. Both numbers shall be added and indicated.

NOTE Detailed instructions for the Device tests are specified in [9].

G.2.2 Test of a Master

The Device AUX 1 (see Figure G.1) shall use M-sequence TYPE_2_5. Its input Process Data shall be generated by an 8 bit random or pseudo random generator. The Master shall copy the input Process Data of any received Device message to the output Process Data of the next Master message to be sent. The cycle time shall be according to Table G.1. The Device AUX 1 shall compare the output Process Data with the previously sent input Process Data and count the number of deviations. The Device shall also count the number of missing (not received within the expected cycle time) or received perturbed Master messages. All numbers shall be added and indicated.

NOTE 1 A deviation of sent and received Process Data indicates to the AUX1 that the EUT (Master) did not receive the Device message.

NOTE 2 Detailed instructions for the Master tests are specified in [9].

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex H
(informative)

Residual error probabilities

H.1 Residual error probability of the SDCI data integrity mechanism

Figure H.1 shows the residual error probability (REP) of the SDCI data integrity mechanism consisting of the checksum data integrity procedure ("XOR6") as specified in A.1.6 and the UART parity. The diagram refers to IEC 60870-5-1 with its data integrity class I2 for a minimum Hamming distance of 4 (red dotted line).

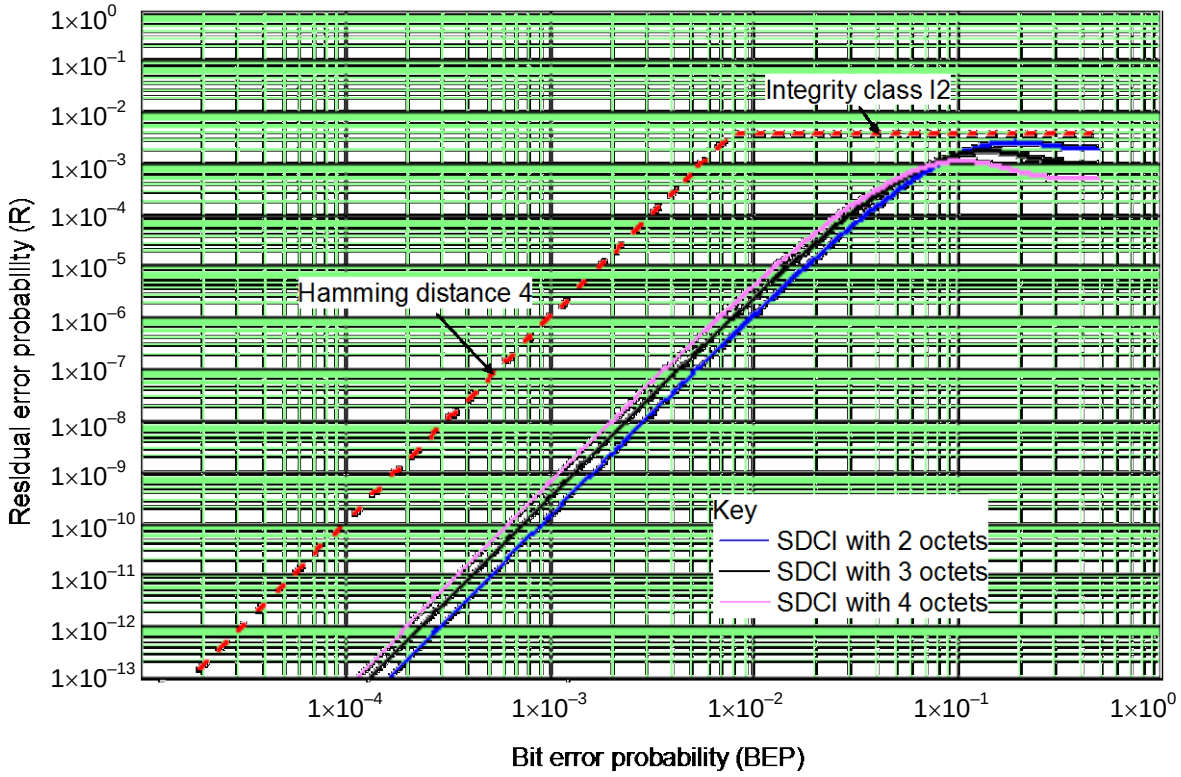


Figure H.1 – Residual error probability for the SDCI data integrity mechanism

The blue line shows the residual error curve for a data length of 2 octets. The black curve shows the residual error curve for a data length of 3 octets. The purple curve shows the residual error curve for a data length of 4 octets.

H.2 Derivation of EMC test conditions

The performance criterion A in G.1.3 is derived from requirements specified in IEC 61158-2 in respect to interference susceptibility and error rates (citation; "*frames*" translates into "messages" within this standard):

- *Only 1 undetected erroneous frame in 20 years at 1 600 frames/s;*
- *The ratio of undetected to detected frames shall not exceed 10^{-6};*
- *EMC tests shall not show more than 6 erroneous frames within 100 000 frames.*

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

With SDCI, the first requirement transforms into the Equation (H.1). This equation allows determining a value of BEP. The equation can be resolved in a numerical way.

$$F20 \times R(BEP) \leq 1 \quad \text{(H.1)}$$

where

F20 is the number of messages in 20 years;

R(BEP) is the residual error probability of the checksum and parity mechanism (Figure H.1);

BEP is the bit error probability from Figure H.1.

The objective of the EMC test is to proof that the BEP of the SDCI communication meets the value determined in the first step. The maximum number of detected perturbed messages is chosen to be 6 here for practical reasons. The number of required SDCI test messages can be determined with the help of Equation (H.2) and the value of BEP determined in the first step.

$$NoTF \geq \frac{1}{BEP} \times \frac{1}{BitPerF} \times NopErr \quad \text{(H.2)}$$

where

NoTF is the number of test messages;

BitPerF is the number of bit per message;

NopErr is the maximum number of detected perturbed messages = 6.

Equation (H.2) is only valid under the assumption that messages with 1 bit error are more frequent than messages with more bit errors. An M-sequence consists of two messages. Therefore, the calculated number of test messages has to be divided by 2 to provide the numbers of M-sequences for Table G.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex I
(informative)

Example sequence of an ISDU transmission

Figure I.1 demonstrates an example for the transmission of ISDUs using an AL_Read service with a 16-bit Index and Subindex for 19 octets of user data with mapping to an M-sequence TYPE_2_5 for sensors and with interruption in case of an Event transmission.

Master								Device			
		FC		CKT		PD	OD	OD	PD	CKS	
comment (state, action) (see in Table 46)	cycle nr	R W 1bit	Com Chan. 2bit	Flow CTRL 5bit	Frame Typ 2bit	CHK 6bit	Process Data 8bit	OnReq Data Master 8bit	OnReq Data Device 8bit	Process Data	CHK E PD	comment (state, action)
Idle_1	0	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	OnReq idle
ISDURequest_2, transmission,	1	0111	0000	10	xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	2	0110	0001	10	xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	3	0110	0010	10	xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	4	0110	0011	10	xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	5	0110	0100	10	xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	6	1111	0000	10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	7	1111	0000	10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	8	1111	0000	10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	9	1111	0000	10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	10	1111	0000	10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	11	1111	0000	10	xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	12	1110	0001	10	xxxxxx	xxxxxxxx		0001 0011	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	13	1110	0010	10	xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	14	1110	0011	10	xxxxxx	xxxxxxxx		Data 2	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	15	1110	0100	10	xxxxxx	xxxxxxxx		Data 3	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	16	1110	0101	10	xxxxxx	xxxxxxxx		Data 4	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	17	1110	0110	10	xxxxxx	xxxxxxxx		Data 5	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	18	1110	0111	10	xxxxxx	xxxxxxxx		Data 6	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	19	1110	1000	10	xxxxxx	xxxxxxxx		Data 7	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, no response, retry in next cycle	20	1110	1001	10	Err	xxxxxxxx				xxxxxx	ISDUResponse_4, korrupted CHK, don' t send resp.
ISDUResponse_4, no response, retry in next cycle	21	1110	1001	10	Err	xxxxxxxx				xxxxxx	ISDUResponse_4, corrupted CHK, don' t send resp.
ISDUResponse_4, reception	22	1110	1001	10	xxxxxx	xxxxxxxx		Data 8	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	34	1110	1010	10	xxxxxx	xxxxxxxx		Data 9	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception, start eventhandler	35	1110	1011	10	xxxxxx	xxxxxxxx		Data 10	xxxxxxxx	**1** 0 xxxxxx	ISDUResponse_4, transmission, freeze event
Read_Event_2, reception	36	1100	0000	10	xxxxxx	xxxxxxxx		Diag State with detail	xxxxxxxx	**1** 0 xxxxxx	Read_Event_2, transmission
Read_Event_2, reception	37	110x	xxxx	10	xxxxxx	xxxxxxxx		Event qualifier	xxxxxxxx	**1** 0 xxxxxx	Read_Event_2, transmission
Command handler_2, transmission set PDOutdata state to invalid	38	0010	0000	10	xxxxxx	xxxxxxxx	1001 1001		xxxxxxxx	**1** 0 xxxxxx	ComandHandler_2, reception, set PDOutdata state to invalid
Read_Event_2, reception	39	110x	xxxx	10	xxxxxx	xxxxxxxx		ErrorCode msb	xxxxxxxx	**1** 0 xxxxxx	Read_Event_2, transmission
Read_Event_2, reception	40	110x	xxxx	10	xxxxxx	xxxxxxxx		ErrorCode lsb	xxxxxxxx	**1** 0 xxxxxx	Read_Event_2, transmission
EventConfirmation_4, transmission, event handler idle	41	0100	0000	10	xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	**0** 0 xxxxxx	EventConfirmation, reception
ISDUResponse_4, reception	42	1110	1100	10	xxxxxx	xxxxxxxx		Data 11	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	43	1110	1101	10	xxxxxx	xxxxxxxx		Data 12	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	44	1110	1110	10	xxxxxx	xxxxxxxx		Data 13	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	45	1110	1111	10	xxxxxx	xxxxxxxx		Data 14	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	46	1110	0000	10	xxxxxx	xxxxxxxx		Data 15	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	47	1110	0001	10	xxxxxx	xxxxxxxx		Data 16	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	48	1110	0010	10	xxxxxx	xxxxxxxx		CHKPDU	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
Idle_1	49	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	50	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	51	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	52	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Write Parameter, transmission	53	0011	0000	10	xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	0 0 xxxxxx	Write Parameter, reception
Read Parameter, reception	54	1011	0000	10	xxxxxx	xxxxxxxx		xxxxxxxx	xxxxxxxx	0 0 xxxxxx	Read Parameter, transmission
Idle_1	55	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	56	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	57	1111	0001	10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1

Figure I.1 – Example for ISDU transmissions *(1 of 2)*

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ISDURequest_2, transmission	58	0111 0000	10 xxxxxx	xxxxxxxx	0001 1011		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	59	0110 0001	10 xxxxxx	xxxxxxxx	Index		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	60	0110 0010	10 xxxxxx	xxxxxxxx	Data 1		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	61	0110 0011	10 xxxxxx	xxxxxxxx	Data 2		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	62	0110 0100	10 xxxxxx	xxxxxxxx	Data 3		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	63	0110 0101	10 xxxxxx	xxxxxxxx	Data 4		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	64	0110 0110	10 xxxxxx	xxxxxxxx	Data 5		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	65	0110 0111	10 xxxxxx	xxxxxxxx	Data 6		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	66	0110 1000	10 xxxxxx	xxxxxxxx	Data 7		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	67	0110 1001	10 xxxxxx	xxxxxxxx	Data 8		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	68	0110 1010	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	69	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	70	1111 0000	10 xxxxxx	xxxxxxxx		0101 0010	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	71	1110 0001	10 xxxxxx	xxxxxxxx		CHKPDU	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
Idle_1	72	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	73	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
ISDURequest_2, transmission,	74	0111 0000	10 xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	75	0110 0001	10 xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	76	0110 0010	10 xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	77	0110 0011	10 xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	78	0110 0100	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	79	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	80	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	81	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	82	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	83	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	84	1111 0000	10 xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	85	1110 0001	10 xxxxxx	xxxxxxxx		0001 1110	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	86	1110 0010	10 xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, ABORT	87	1111 1111	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, ABORT
Idle_1	88	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	89	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1

Figure I.1 *(2 of 2)*

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annex J
(informative)

Recommended methods for detecting parameter changes

J.1 CRC signature

Cyclic Redundancy Checking belongs to the HASH function family. A CRC signature across all changeable parameters can be calculated by the Device with the help of a so-called proper generator polynomial. The calculation results in a different signature whenever the parameter set has been changed. It should be noted that the signature secures also the octet order within the parameter set. Any change in the order when calculating the signature will lead to a different value. The quality of securing (undetected changes) depends heavily on both the CRC generator polynomial and the length (number of octets) of the parameter set. The seed value should be > 0. One calculation method uses directly the formula, another one uses octet shifting and lookup tables. The first one requests less program memory and is a bit slower, the other one requires memory for a lookup table (1×2^{10} octets for a 32 bit signature) and is fast. The parameter data set comparison is performed in state "Checksum_9" of the Data Storage (DS) state machine in Figure 100. Table J.1 lists several possible generator polynomials and their detection level.

Table J.1 – Proper CRC generator polynomials

Generator polynomial	Signature	Data length	Undetected changes
0x9B	8 bit	1 octet	$< 2^{-8}$ (not recommended)
0x4EAB	16 bit	1 < octets < 3	$< 2^{-16}$ (not recommended)
0x5D6DCB	24 bit	1 < octets < 4	$< 2^{-24}$ (not recommended)
0xF4ACFB13	32 bit	1 < octets < 2^{32}	$< 2^{-32}$ (recommended)

J.2 Revision counter

A 32 bit revision counter can be implemented, counting any change of the parameter set. The Device shall use a random initial value for the Revision Counter. The counter itself shall not be stored via Index List of the Device. After the download the actual counter value is read back from the Device to avoid multiple writing initiated by the download sequence. The parameter data set comparison is performed in state "Checksum_9" of the Data Storage (DS) state machine in Figure 100.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Bibliography

[1] IEC 60050 (all parts), *International Electrotechnical Vocabulary* (available at <http://www.electropedia.org>)

[2] IEC 60870-5-1:1990, *Telecontrol equipment and systems – Part 5: Transmission protocols – Section One: Transmission frame formats*

[3] IEC 61158-2, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

[4] IEC/TR 62453-61, *Field device tool interface specification – Part 61: Device Type Manager (DTM) Styleguide for common object model*

[5] ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

[6] IO-Link Consortium, *IO Device Description (IODD), V1.1, Order No. 10.012* (available at <http://www.io-link.com>)

[7] IO-Link Consortium, *IO-Link Smart Sensor Profile, V1.1, Order No. 10.042* (available at <http://www.io-link.com>)

[8] IO-Link Consortium, *IO-Link Communication, V1.0, January 2009, Order No. 10.002* (available at <http://www.io-link.com>)

[9] IO-Link Consortium, *IO-Link Test Specification, V1.1, Order No. 10.032* (available at <http://www.io-link.com>)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

SOMMAIRE

AVANT-PROPOS 273
INTRODUCTION 275
1 Domaine d'application 278
2 Références normatives 278
3 Termes, définitions, symboles, abréviations et conventions 280
3.1 Termes et définitions 280
3.2 Symboles et abréviations 283
3.3 Conventions 285
3.3.1 Généralités 285
3.3.2 Paramètres de services 285
3.3.3 Procédures de services 286
3.3.4 Attributs de services 286
3.3.5 Figures 287
3.3.6 Ordre de transmission des octets 287
3.3.7 Descriptifs comportementaux 288
4 Aperçu global de l'interface SDCI (IO-Link™) 289
4.1 Objectif de la technologie 289
4.2 Positionnement dans la hiérarchie de l'automatisation 289
4.3 Câblage, connecteurs et alimentation 291
4.4 Fonctionnalités de communication de l'interface SDCI 291
4.5 Rôle d'un Maître 295
4.6 Configuration SDCI 295
4.7 Mise en correspondance avec les bus de terrain 296
4.8 Structure de la norme 296
5 Couche Physique (PL) 297
5.1 Généralités 297
5.1.1 Principes de base 297
5.1.2 Topologie 297
5.2 Services de Couche Physique 298
5.2.1 Aperçu 298
5.2.2 Services de PL 300
5.3 Émetteur/Récepteur 301
5.3.1 Méthode de description 301
5.3.2 Exigences électriques 302
5.3.3 Exigences temporelles 307
5.4 Alimentation 311
5.4.1 Options d'alimentation 311
5.4.2 Exigences de mise sous tension 312
5.5 Support 313
5.5.1 Connecteurs 313
5.5.2 Câblage 315
6 Entrées et sorties normalisées (SIO) 316
7 Couche de liaison de données (DL) 316
7.1 Généralités 316
7.2 Services de la couche liaison de données 318
7.2.1 Services de la section B de couche DL 318

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.2 Services de la section A de couche DL 330
7.3 Protocole de la couche liaison de données 335
7.3.1 Aperçu 335
7.3.2 Module de traitement de mode DL 336
7.3.3 Module de traitement des messages 347
7.3.4 Module de traitement des Données de Processus 356
7.3.5 Module de traitement de Données à la Demande 360
7.3.6 Module de traitement des ISDU 364
7.3.7 Module de traitement des commandes 369
7.3.8 Module de traitement d'Événements 372
8 Couche d'application (AL) 376
8.1 Généralités 376
8.2 Services de la Couche d'Application 377
8.2.1 Services AL au sein du Maître et du Dispositif 377
8.2.2 Services d'AL 378
8.3 Protocole de couche d'application 386
8.3.1 Aperçu 386
8.3.2 Transfert de Données à la Demande 387
8.3.3 Traitement des Événements 393
8.3.4 Cycles de Données de Processus 398
9 Gestion système (SM) 400
9.1 Généralités 400
9.2 Gestion Système du Maître 400
9.2.1 Aperçu 400
9.2.2 Services SM du Maître 404
9.2.3 Protocole SM du Maître 409
9.3 Gestion Système du Dispositif 417
9.3.1 Aperçu 417
9.3.2 Services SM du Dispositif 421
9.3.3 Protocole SM du Dispositif 427
10 Dispositif 435
10.1 Aperçu 435
10.2 Échange de Données de Processus (PDE) 436
10.3 Gestionnaire de Paramètres (PM) 437
10.3.1 Généralités 437
10.3.2 Diagramme d'états du Gestionnaire de Paramètres 437
10.3.3 Paramètre dynamique 439
10.3.4 Paramètre unique 440
10.3.5 Paramètre de bloc 441
10.3.6 Accès concurrent au paramétrage 444
10.3.7 Traitement des commandes 444
10.4 Stockage des Données (DS) 444
10.4.1 Généralités 444
10.4.2 Diagramme d'états du Stockage des Données 445
10.4.3 Configuration du DS 447
10.4.4 Espace mémoire du DS 447
10.4.5 Index_List de DS 448
10.4.6 Disponibilité des paramètres de DS 448
10.4.7 DS sans ISDU 448

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.4.8 Indication de modification de paramètres de DS 448
10.5 Répartiteur d'Événements (ED) 448
10.6 Fonctionnalités du Dispositif 449
10.6.1 Généralités 449
10.6.2 Rétrocompatibilité d'un Dispositif 449
10.6.3 Compatibilité de versions du protocole 449
10.6.4 Réglages d'usine 449
10.6.5 Réinitialisation d'application 449
10.6.6 Réinitialisation d'un Dispositif 450
10.6.7 Indication visuelle d'état de l'interface SDCI 450
10.6.8 Verrouillage de l'accès aux paramètres 450
10.6.9 Verrouillage du Stockage des Données 450
10.6.10 Verrouillage de paramètres du Dispositif 450
10.6.11 Verrouillage de l'interface utilisateur d'un Dispositif 450
10.6.12 Décalage temporel 450
10.6.13 Concept de Stockage des Données 451
10.6.14 Paramètre de bloc 451
10.7 Règles et contraintes de conception d'un Dispositif 451
10.7.1 Généralités 451
10.7.2 Données de Processus 452
10.7.3 Perte de communication 452
10.7.4 Paramètre direct 452
10.7.5 Canal de communication des ISDU 452
10.7.6 Règles de DeviceID relatives aux variantes de Dispositifs 453
10.7.7 Constantes de protocole 453
10.8 Descriptif de Dispositif d'E/S (IODD) 453
10.9 Diagnostic d'un Dispositif 454
10.9.1 Concepts 454
10.9.2 Événements 455
10.9.3 Indicateurs visuels 457
10.10 Connectivité d'un Dispositif 457
11 Maître 457
11.1 Aperçu 457
11.1.1 Modèle générique d'intégration système d'un Maître 457
11.1.2 Structure et services d'un Maître 459
11.2 Gestionnaire de configuration (CM) 462
11.2.1 Généralités 462
11.2.2 Paramètres de Configuration 464
11.2.3 Diagramme d'états du Gestionnaire de Configuration 466
11.3 Stockage des Données(DS) 469
11.3.1 Aperçu 469
11.3.2 Objet de données DS 469
11.3.3 Diagramme d'états DS 469
11.3.4 Sélection de paramètres pour Stockage des Données 478
11.4 Échange de Données à la Demande (ODE) 478
11.5 Unité de Diagnostic (DU) 479
11.6 Échange de PD (PDE) 482
11.6.1 Généralités 482
11.6.2 Mise en correspondance des Données de Processus 482

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.6.3 État du qualificatif non valide/valide des Données de Processus 483
11.7 Outil de configuration des ports et Dispositifs (PDCT) 484
11.7.1 Généralités 484
11.7.2 Exemples de présentation de base 484
11.8 Application passerelle 486
11.8.1 Généralités 486
11.8.2 Modification de la configuration du Dispositif, y compris le Stockage des Données 486
11.8.3 Serveur de paramètres et contrôle des recettes 486
11.8.4 Paramètres anonymes 486
11.8.5 Mode port virtuel DIwithSDCI 487
Annexe A (normative) Codages, contraintes temporelles et erreurs 490
Annexe B (normative) Paramètres et commandes 517
Annexe C (normative) ErrorTypes (erreurs d'ISDU) 537
Annexe D (normative) EventCodes (informations de diagnostic) 542
Annexe E (normative) Types de Données 545
Annexe F (normative) Structure de l'objet de données "Stockage de Données" 557
Annexe G (normative) Conformité du Maître et du Dispositif 558
Annexe H (informative) Probabilités d'erreurs résiduelles 566
Annexe I (informative) Exemple d'une séquence de transmission d'ISDU 568
Annexe J (informative) Méthodes recommandées pour la détection de modifications de paramètres 570
Bibliographie 571

Figure 1 – Exemple de service confirmé 287
Figure 2 – Ordre de stockage en mémoire et de transmission pour des types de données à base de MOTS 288
Figure 3 – Compatibilité de la SDCI avec la CEI 61131-2 289
Figure 4 – Domaine de la technologie SDCI au sein de la hiérarchie de l'automatisation 290
Figure 5 – Modèle de Dispositif générique pour SDCI (vue du côté Maître) 292
Figure 6 – Relation entre nature des données et types de transmission 293
Figure 7 – Transfert d'objets au niveau de la couche d'application (AL) 295
Figure 8 – Structure logique du Maître et du Dispositif 296
Figure 9 – Système de connexion à trois conducteurs 297
Figure 10 – Topologie de l'interface SDCI 298
Figure 11 – Couche physique (Maître) 298
Figure 12 – Couche physique (Dispositif) 299
Figure 13 – Schéma de référence du circuit d'attaque de la ligne 302
Figure 14 – Schéma de référence du récepteur 302
Figure 15 – Schéma de référence pour le système de connexion à 3 conducteurs SDCI 303
Figure 16 – Définition des niveaux de tension 304
Figure 17 – Seuils de commutation 305
Figure 18 – Format d'une trame UART de l'interface SDCI 307
Figure 19 – Diagramme en oeil pour la détection de 'H' et 'L' 308
Figure 20 – Diagramme en oeil pour la détection correcte d'une trame UART 309
Figure 21 – Demande de réactivation 311

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 22 – Temporisation de la mise sous tension pour Power1 312
Figure 23 – Vue de face de l'agencement des broches 314
Figure 24 – Définitions des ports de classes A et B 315
Figure 25 – Schéma de référence de la capacité effective de la ligne et de la résistance de boucle 315
Figure 26 – Structure et services de la couche de liaison de données (Maître) 317
Figure 27 – Structure et services de la couche de liaison de données (Dispositif) 318
Figure 28 – Diagrammes d'états de la couche de liaison de données 336
Figure 29 – Exemple d'une tentative d'établissement de communication 337
Figure 30 – Échec d'une tentative d'établissement de communication 338
Figure 31 – Stratégie de nouvelle tentative d'établissement de communication 338
Figure 32 – Procédure de repli 340
Figure 33 – Diagramme d'états du module de traitement de mode DL du Maître 341
Figure 34 – Sous-machine 1 d'établissement de la communication 342
Figure 35 – Diagramme d'états du module de traitement de mode DL du Dispositif 345
Figure 36 – Séquences de messages SDCI 348
Figure 37 – Aperçu des types de séquence M 349
Figure 38 – Diagramme d'états du module de traitement des messages du Maître 350
Figure 39 – Sous-machine "Response 3" du module de traitement des messages 351
Figure 40 – Sous-machine "Response 8" du module de traitement des messages 351
Figure 41 – Sous-machine "Response 15" du module de traitement des messages 352
Figure 42 – Diagramme d'états du module de traitement des messages du Dispositif 355
Figure 43 – Mode d'entrelacement pour la transmission segmentée des Données de Processus 357
Figure 44 – Diagramme d'états du module de traitement des Données de Processus du Maître 358
Figure 45 – Diagramme d'états du module de traitement des Données de Processus du Dispositif 359
Figure 46 – Diagramme d'états du module de traitement de Données à la Demande du Maître 361
Figure 47 – Diagramme d'états du module de traitement de Données à la Demande du Dispositif 363
Figure 48 – Structure de l'ISDU 364
Figure 49 – Diagramme d'états du module de traitement des ISDU du Maître 366
Figure 50 – Diagramme d'états du module de traitement des ISDU du Dispositif 368
Figure 51 – Diagramme d'états du module de traitement des commandes du Maître 370
Figure 52 – Diagramme d'états du module de traitement des commandes du Dispositif 371
Figure 53 – Diagramme d'états du module de traitement d'Événements du Maître 373
Figure 54 – Diagramme d'états du module de traitement d'Événements du Dispositif 375
Figure 55 – Structure et services de la couche d'application (Maître) 376
Figure 56 – Structure et services de la couche d'application (Dispositif) 377
Figure 57 – Diagramme d'états d'OD de la couche AL du Maître 387
Figure 58 – Diagramme d'états d'OD de la couche AL du Dispositif 389
Figure 59 – Diagramme de séquence pour la transmission de Données à la Demande 391

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 60 – Diagramme de séquence pour des Données à la Demande en cas d'erreurs 392
Figure 61 – Diagramme de séquence pour des Données à la Demande en cas d'expiration de délai 393
Figure 62 – Diagramme d'états d'Événements de la couche AL du Maître 394
Figure 63 – Diagramme d'états d'Événements de la couche AL du Dispositif 395
Figure 64 – Planification d'Événements uniques 397
Figure 65 – Diagramme de séquence pour des Données de Processus de sortie 399
Figure 66 – Diagramme de séquence pour des Données de Processus d'entrée 400
Figure 67 – Structure et services de la Gestion Système du Maître 401
Figure 68 – Organigramme séquentiel du cas d'utilisation "Configuration de port x" 404
Figure 69 – Diagramme d'états principal de la Gestion Système du Maître 410
Figure 70 – Sous-machine de la SM du Maître CheckCompatibility_1 412
Figure 71 – Activité pour l'état "CheckVxy" 414
Figure 72 – Activité pour l'état "CheckCompV10" 414
Figure 73 – Activité pour l'état "CheckComp" 415
Figure 74 – Activité (écriture de paramètres) dans l'état "RestartDevice" 415
Figure 75 – Sous-machine de la SM du Maître CheckSerNum_3 416
Figure 76 – Activité (vérification du SerialNumber) pour l'état CheckSerNum_3 417
Figure 77 – Structure et services de la Gestion Système (Dispositif) 418
Figure 78 – Organigramme séquentiel du cas d'utilisation "INACTIVE – SIO – SDCI – SIO" 420
Figure 79 – Diagramme d'états de la Gestion Système du Dispositif 428
Figure 80 – Organigramme séquentiel de démarrage ordinaire du Dispositif 431
Figure 81 – Organigramme séquentiel du démarrage d'un Dispositif en mode compatibilité 433
Figure 82 – Organigramme séquentiel du démarrage d'un Dispositif en cas d'échec de la compatibilité 434
Figure 83 – Structure et services d'un Dispositif 436
Figure 84 – Diagramme d'états du Gestionnaire de Paramètres (PM) 438
Figure 85 – Résultat positif et négatif d'une vérification de paramètre 440
Figure 86 – Téléchargement aval positif de paramètre de bloc avec demande de Stockage des Données 442
Figure 87 – Téléchargement aval négatif de paramètre de bloc 443
Figure 88 – Diagramme d'états du Stockage des Données (DS) 445
Figure 89 – Séquence du message de demande de Stockage des Données 447
Figure 90 – Synchronisation du cycle 451
Figure 91 – Flux d'Événements en cas d'erreurs successives 456
Figure 92 – Temporisation de l'indicateur à DEL du Dispositif 457
Figure 93 – Relation générique entre technologie SDCI et technologie de bus de terrain 458
Figure 94 – Structure et services d'un Maître 461
Figure 95 – Relations entre les applications communes d'un Maître 461
Figure 96 – Diagramme de séquence des actions du Gestionnaire de Configuration 463
Figure 97 – Ports en mode MessageSync 465

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure 98 – Diagramme d'états du Gestionnaire de Configuration 467
Figure 99 – Principal diagramme d'états du mécanisme de Stockage des Données 470
Figure 100 – Sous-machine "UpDownload_2" du mécanisme de Stockage des Données 471
Figure 101 – Sous-machine du Stockage des Données"Upload_7" 472
Figure 102 – Diagramme de séquence du téléchargement amont de Stockage des Données 473
Figure 103 – Sous-machine du Stockage des Données "Download_10" 474
Figure 104 – Diagramme de séquence du téléchargement aval de Stockage des Données 475
Figure 105 – Diagramme d'états de l'Échange de Données à la Demande 479
Figure 106 – Vue système de la propagation des informations de diagnostic SDCI via des Événements 482
Figure 107 – Mise en correspondance des Données de Processus entre ports et flux de données de passerelle 482
Figure 108 – Propagation de l'état qualificatif de PD entre Maître et Dispositif 483
Figure 109 – Exemple 1 de présentation de l'affichage d'un PDCT 485
Figure 110 – Exemple 2 de présentation de l'affichage d'un PDCT 486
Figure 111 – Configuration alternative de Dispositif 487
Figure 112 – Mode de port virtuel "DIwithSDCI" 488
Figure A.1 – Contrôle de séquence M 490
Figure A.2 – Octet de Somme de contrôle/Type de séquence M 491
Figure A.3 – Octet de Somme de contrôle/État 492
Figure A.4 – Principe de calcul et de compression de la somme de contrôle 494
Figure A.5 – Séquence M de TYPE_0 495
Figure A.6 – Séquence M de TYPE_1_1 495
Figure A.7 – Séquence M de TYPE_1_2 496
Figure A.8 – Séquence M de TYPE_1_V 496
Figure A.9 – Séquence M de TYPE_2_1 497
Figure A.10 – Séquence M de TYPE_2_2 498
Figure A.11 – Séquence M de TYPE_2_3 498
Figure A.12 – Séquence M de TYPE_2_4 499
Figure A.13 – Séquence M de TYPE_2_5 499
Figure A.14 – Séquence M de TYPE_2_6 500
Figure A.15 – Séquence M de TYPE_2_V 500
Figure A.16 – Temporisation de séquence M 504
Figure A.17 – Octet de "Service I" 506
Figure A.18 – Vérification de l'intégrité des ISDU via CHKPDU 509
Figure A.19 – Exemples de formats de demande d'ISDU 510
Figure A.20 – Exemples de ISDU de réponse 511
Figure A.21 – Exemples de ISDU de demande de lecture et d'écriture 512
Figure A.22 – Structure du StatusCode de type 1 513
Figure A.23 – Structure du StatusCode de type 2 514
Figure A.24 – Indication d'Événements activés 514
Figure A.25 – Structure du EventQualifier 515

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Figure B.1 – Classification et mise en correspondance des Paramètres Directs 517
Figure B.2 – MinCycleTime 520
Figure B.3 – Capacité de séquence M 521
Figure B.4 – RevisionID 521
Figure B.5 – ProcessDataIn 522
Figure B.6 – Espace d'index pour des objets de données ISDU 525
Figure B.7 – Structure du Décalage temporel 535
Figure E.1 – Exemples de codage d'UIntegerT 546
Figure E.2 – Exemples de codage de IntegerT 548
Figure E.3 – Accès singulier de StringT 550
Figure E.4 – Exemple de codage de OctetStringT 550
Figure E.5 – Définition de TimeT 551
Figure E.6 – Exemple d'une structure de données ArrayT 553
Figure E.7 – Deuxième exemple (2) d'une structure de RecordT 554
Figure E.8 – Troisième exemple (3) d'une structure de RecordT 555
Figure E.9 – Demandes d'écriture pour l'exemple 3 556
Figure G.1 – Montage d'essai de décharges électrostatiques (Maître) 561
Figure G.2 – Montage d'essai de champ électromagnétique aux fréquences radioélectriques (Maître) 561
Figure G.3 – Montage d'essai de transitoires rapides (Maître) 562
Figure G.4 – Montage d'essai de fréquences radioélectriques en mode commun (Maître) 562
Figure G.5 – Montage d'essai de décharges électrostatiques (Dispositif) 563
Figure G.6 – Montage d'essai de champ électromagnétique aux fréquences radioélectriques (Dispositif) 564
Figure G.7 – Montage d'essai de transitoires rapides (Dispositif) 564
Figure G.8 – Montage d'essai de fréquences radioélectriques en mode commun (Dispositif) 565
Figure H.1 – Probabilité d'erreurs résiduelles du mécanisme d'intégrité des données SDCI 566
Figure I.1 – Exemple de transmissions d'ISDU *(1 de 2)* 568

Tableau 1 – Attributions des services du Maître et du Dispositif 300
Tableau 2 – PL_SetMode 300
Tableau 3 – PL_WakeUp 301
Tableau 4 – PL_Transfer 301
Tableau 5 – Caractéristiques électriques d'un récepteur 304
Tableau 6 – Caractéristiques électriques d'un port du Maître 305
Tableau 7 – Caractéristiques électriques d'un Dispositif 306
Tableau 8 – Caractéristiques dynamiques de la transmission 309
Tableau 9 – Caractéristiques de la demande de réactivation 311
Tableau 10 – Temporisation de la mise sous tension 312
Tableau 11 – Affectation des broches 313
Tableau 12 – Caractéristiques du câblage 315
Tableau 13 – Affectation des conducteurs du câble 315

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 14 – Attributions des services au sein du Maître et du Dispositif 319
Tableau 15 – DL_ReadParam 319
Tableau 16 – DL_WriteParam 320
Tableau 17 – DL_Read 321
Tableau 18 – DL_Write 322
Tableau 19 – DL_ISDUTransport 323
Tableau 20 – DL_ISDUAbort 324
Tableau 21 – DL_PDOutputUpdate 324
Tableau 22 – DL_PDOutputTransport 325
Tableau 23 – DL_PDInputUpdate 325
Tableau 24 – DL_PDInputTransport 326
Tableau 25 – DL_PDCycle 326
Tableau 26 – DL_SetMode 327
Tableau 27 – DL_Mode 328
Tableau 28 – DL_Event 328
Tableau 29 – DL_EventConf 329
Tableau 30 – DL_EventTrigger 329
Tableau 31 – DL_Control 329
Tableau 32 – Services DL-A au sein du Maître et du Dispositif 330
Tableau 33 – OD 331
Tableau 34 – PD 332
Tableau 35 – EventFlag 333
Tableau 36 – PDInStatus 333
Tableau 37 – MHInfo 334
Tableau 38 – ODTrig 334
Tableau 39 – PDTrig 335
Tableau 40 – Caractéristiques des procédures et des nouvelles tentatives de réactivation 339
Tableau 41 – Caractéristiques temporelles du repli 340
Tableau 42 – Tables de transitions d'états du module de traitement de mode DL du Maître 342
Tableau 43 – Tables de transitions d'états du module de traitement de mode DL du Dispositif 346
Tableau 44 – Table de transitions d'états du module de traitement des messages du Maître 352
Tableau 45 – Tables de transitions d'états du module de traitement des messages du Dispositif 356
Tableau 46 – Tables de transitions d'états du module de traitement des Données de Processus du Maître 358
Tableau 47 – Tables de transitions d'états du module de traitement des Données de Processus du Dispositif 360
Tableau 48 – Tables de transitions d'états du module de traitement des Données à la Demande du Maître 361
Tableau 49 – Tables de transitions d'états du module de traitement des Données à la Demande du Dispositif 363
Tableau 50 – Définitions du FlowCTRL 365

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 51 – Tables de transitions d'états du module de traitement des ISDU du Maître 366
Tableau 52 – Tables de transitions d'états du module de traitement des ISDU du Dispositif 368
Tableau 53 – Codes de contrôle 369
Tableau 54 – Tables de transitions d'états du module de traitement des commandes du Maître 370
Tableau 55 – Tables de transitions d'états du module de traitement des commandes du Dispositif 371
Tableau 56 – Mémoire d'Événements 372
Tableau 57 – Tables de transitions d'états du module de traitement d'Événements du Maître 374
Tableau 58 – Tables de transitions d'états du module de traitement d'Événements du Dispositif 375
Tableau 59 – Services AL au sein du Maître et du Dispositif 378
Tableau 60 – AL_Read 378
Tableau 61 – AL_Write 379
Tableau 62 – AL_Abort 381
Tableau 63 – AL_GetInput 381
Tableau 64 – AL_NewInput 382
Tableau 65 – AL_SetInput 382
Tableau 66 – AL_PDCycle 383
Tableau 67 – AL_GetOutput 383
Tableau 68 – AL_NewOutput 384
Tableau 69 – AL_SetOutput 384
Tableau 70 – AL_Event 385
Tableau 71 – AL_Control 386
Tableau 72 – États et transitions pour le diagramme d'états d'OD de la couche AL du Maître 388
Tableau 73 – États et transitions pour le diagramme d'états d'OD de la couche AL du Dispositif 389
Tableau 74 – États et transitions du diagramme d'états d'Événements de la couche AL du Maître 394
Tableau 75 – États et transitions du diagramme d'états d'Événements de la couche AL du Dispositif 396
Tableau 76 – Services SM au sein du Maître 404
Tableau 77 – SM_SetPortConfig 405
Tableau 78 – Définition de InspectionLevel (IL) 406
Tableau 79 – Définitions des Modes Cibles 406
Tableau 80 – SM_GetPortConfig 407
Tableau 81 – SM_PortMode 408
Tableau 82 – SM_Operate 409
Tableau 83 – Tables de transitions d'états de la Gestion Système du Maître 410
Tableau 84 – Tables de transitions d'états de la sous-machine du Maître CheckCompatibility_1 412
Tableau 85 – Table de transitions d'états de la sous-machine du Maître CheckSerNum_3 416

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 86 – Services SM au sein du Dispositif 421
Tableau 87 – SM_SetDeviceCom 421
Tableau 88 – SM_GetDeviceCom 423
Tableau 89 – SM_SetDeviceIdent 424
Tableau 90 – SM_GetDeviceIdent 425
Tableau 91 – SM_SetDeviceMode 426
Tableau 92 – SM_DeviceMode 426
Tableau 93 – Table de transitions d'états de la Gestion Système du Dispositif 428
Tableau 94 – Tables de transitions d'états du diagramme d'états du PM 438
Tableau 95 – Définitions des vérifications de paramètre 441
Tableau 96 – Table de transitions d'états du diagramme d'états de Stockage des Données 446
Tableau 97 – Aperçu des constantes de protocole pour les Dispositifs 453
Tableau 98 – Classification d'incidents de diagnostiques de Dispositifs 455
Tableau 99 – Temporisation pour des indicateurs à DEL 457
Tableau 100 – Variables internes et Événements de contrôle des applications communes d'un Maître 462
Tableau 101 – Tables de transitions d'états du Gestionnaire de Configuration 468
Tableau 102 – États et transitions des diagrammes d'états de Stockage des Données 476
Tableau 103 – Table de transitions d'états du diagramme d'états de l'ODE 479
Tableau 104 – Transitions d'états du diagramme d'états "DIwithSDCI" 489
Tableau A.1 – Valeurs du canal de communication 491
Tableau A.2 – Valeurs de R/W 491
Tableau A.3 – Valeurs de types de séquence M 492
Tableau A.4 – Types de données pour les données utilisateur 492
Tableau A.5 – Valeurs d'état des PD 493
Tableau A.6 – Valeurs du drapeau d'Événement 493
Tableau A.7 – Types de séquence M pour le mode STARTUP 501
Tableau A.8 – Types de séquence M pour le mode PREOPERATE 501
Tableau A.9 – Types de séquence M pour le mode OPERATE (protocole héritage) 501
Tableau A.10 – Types de séquence M pour le mode OPERATE 502
Tableau A.11 – Valeurs recommandées de MinCycleTimes 504
Tableau A.12 – Définition du "Service I" réduit (quartet) 507
Tableau A.13 – Syntaxe des ISDU 507
Tableau A.14 – Définition de la longueur réduite (quartet) et de la longueur étendue d'octet ExtLength 508
Tableau A.15 – Utilisation des formats d'Index 508
Tableau A.16 – Mise en correspondance des EventCodes (type 1) 513
Tableau A.17 – Valeurs de "INSTANCE" 515
Tableau A.18 – Valeurs de "SOURCE" 515
Tableau A.19 – Valeurs de "TYPE" 515
Tableau A.20 – Valeurs de "MODE" 516
Tableau B.1 – Pages 1 et 2 de Paramètres Directs 518
Tableau B.2 – Types de MasterCommands 519

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau B.3 – Valeurs possibles de MasterCycleTime et de MinCycleTime 520
Tableau B.4 – Valeurs de ISDU 521
Tableau B.5 – Valeurs de SIO 522
Tableau B.6 – Combinaisons admises de BYTE (octet) et Length (longueur) 522
Tableau B.7 – Règles de mise en œuvre des paramètres et commandes 525
Tableau B.8 – Attribution des Index d'objets de données (paramètres de Dispositif) 525
Tableau B.9 – Codage d'une SystemCommand (ISDU) 527
Tableau B.10 – Attributions d'Index de Stockage de Données 528
Tableau B.11 – Structure de Index_List 530
Tableau B.12 – Possibilités de verrouillage du Dispositif 531
Tableau B.13 – Paramètre d'état du Dispositif 533
Tableau B.14 – État détaillé du Dispositif (Index 0x0025) 534
Tableau B.15 – Codage de la base de temps et valeurs du décalage temporel 535
Tableau C.1 – ErrorTypes 537
Tableau C.2 – ErrorTypes dérivés 540
Tableau D.1 – EventCodes 542
Tableau D.2 – EventCodes SDCI de base 544
Tableau E.1 – BooleanT 545
Tableau E.2 – Codage BooleanT 545
Tableau E.3 – UIntegerT 546
Tableau E.4 – IntegerT 546
Tableau E.5 – Codage de IntegerT (8 octets) 547
Tableau E.6 – Codage de IntegerT (4 octets) 547
Tableau E.7 – Codage de IntegerT (2 octets) 547
Tableau E.8 – Codage de IntegerT (1 octet) 547
Tableau E.9 – Float32T 548
Tableau E.10 – Codage de Float32T 548
Tableau E.11 – StringT 549
Tableau E.12 – OctetStringT 550
Tableau E.13 – TimeT 551
Tableau E.14 – Codage de TimeT 551
Tableau E.15 – TimeSpanT 552
Tableau E.16 – Codage de TimeSpanT 552
Tableau E.17 – Règles de structuration d'ArrayT 552
Tableau E.18 – Exemple d'accès à un type ArrayT 553
Tableau E.19 – Règles de structuration de RecordT 553
Tableau E.20 – Premier exemple (1) d'accès à un RecordT 554
Tableau E.21 – Deuxième exemple (2) d'accès à un RecordT 554
Tableau E.22 – Troisième exemple (3) d'accès à un RecordT 555
Tableau F.1 – Structure de l'objet de données DS enregistré 557
Tableau F.2 – Informations d'en-tête associées aux objets de données DS enregistrés 557
Tableau G.1 – Conditions d'essai CEM pour l'interface SDCI 559
Tableau G.2 – Niveaux d'essais CEM 559

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau J.1 – Polynômes générateurs de CRC corrects..... 570

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

AUTOMATES PROGRAMMABLES –

Partie 9: Interface de communication numérique point à point pour petits capteurs et actionneurs (SDCI)

AVANT-PROPOS

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

La norme internationale CEI 61131-9 a été établie par le sous-comité 65B: Équipements de mesure et de contrôle-commande, en collaboration avec le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65B/874/FDIS	65B/889/RVD

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

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Une liste de toutes les parties de la série CEI 61131, publiées sous le titre général *Automates programmables*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

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

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

INTRODUCTION

0.1 Généralités

La CEI 61131-9 fait partie d'une série de normes sur les automates programmables et les périphériques associés, et il convient de la lire conjointement avec les autres parties de la série.

En cas de conflit entre cette norme et d'autres normes CEI (à l'exception des normes fondamentales de sécurité), il convient de considérer les dispositions de cette norme qui régissent le domaine des automates programmables et leurs périphériques associés.

L'utilisation accrue des microcontrôleurs intégrés dans des capteurs et actionneurs de faible coût a été l'occasion d'ajouter des données de diagnostic et de configuration permettant de prendre en charge les exigences croissantes des applications.

Le moteur premier de la technologie SDCI (IO-Link™[1]) est la nécessité pour ces capteurs et actionneurs de faible coût d'échanger ces données de diagnostic et de configuration avec un contrôleur (PC ou Automate programmable) en utilisant une technologie de communication numérique peu onéreuse tout en préservant une rétrocompatibilité avec les signaux DI/DO courants.

Au sein des concepts de bus de terrain, la technologie SDCI définit une interface générique pour la connexion de capteurs et d'actionneurs à une unité Maître, qui peut être associée à des fonctionnalités de passerelle pour constituer un nœud déporté d'E/S de bus de terrain.

Tout Dispositif conforme SDCI peut être relié à n'importe quel port disponible de l'interface du Maître. Les Dispositifs conformes SDCI réalisent en leur sein une conversion physique en numérique, puis en communiquent directement le résultat en format normalisé au moyen d'une "commutation codée" de la ligne de signal d'E/S 24 V, rendant ainsi inutile l'utilisation de divers modules DI, DO, AI, AO et d'une multitude de câbles.

La topologie physique est réalisée en point à point de chaque Dispositif vers le Maître au moyen de 3 conducteurs sur des distances allant jusqu'à 20 m. L'interface SDCI physique est rétrocompatible avec la signalisation E/S 24V classique définie dans la CEI 61131-2. Elle permet des vitesses de transmission de 4,8 kbit/s, 38,4 kbit/s et 230,4 kbit/s.

Le Maître de l'interface SDCI détecte, identifie et gère les Dispositifs branchés aux ports de l'interface.

Des outils permettent d'associer les Dispositifs à leurs Descriptifs de Dispositifs d'E/S électroniques (IODD, de l'anglais I/O Device Description) ainsi que leur configuration ultérieure pour qu'ils correspondent aux exigences des applications.

La technologie SDCI définit trois différents niveaux de fonctionnalités de diagnostic: pour une réaction immédiate par des moyens automatisés au cours de la phase de production, pour une réaction à moyen terme par intervention de l'opérateur ou pour la mise en service et la maintenance à long terme sur la base d'informations de diagnostic étendues.

La structure de la présente norme est décrite en 4.8.

---

[1] IO-Link™ est une marque commerciale de "IO-Link Consortium". Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme ne nécessite pas l'utilisation des logotypes déposés pour IO-Link™. L'utilisation des logotypes déposés pour IO-Link™ nécessite l'autorisation du "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Il n'est possible de revendiquer la conformité à la CEI 61131-9 que si les exigences de l'Annexe G sont satisfaites.

Les termes d'ordre général sont définis dans la CEI 61131-1 ou dans la série CEI 60050. Les termes plus spécifiques sont définis dans chacune des parties de la norme.

0.2 Déclaration de propriété

La Commission Électrotechnique Internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation d'un brevet intéressant l'interface de communication série point à point pour petits capteurs et actionneurs tels que définis ci-après, la notation [xx] désignant le détenteur des droits de propriété:

DE 10030845B4 EP 1168271B1 US 6889282B2	[AB]	Fieldbus connecting system for actuators or sensors
EP 1203933 B1	[FE]	Sensor device for measuring at least one variable
DE 10 2004 035 831.1	[SI]	Operational status of a computer system is checked by comparison of actual parameters with reference values and modification to software if needed
DE 102 119 39 A1 US 2003/0200323 A1	[SK]	Coupling apparatus for the coupling of devices to a bus system

La CEI ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

Les détenteurs de ces droits de propriété ont donné l'assurance à la CEI qu'ils consentent à négocier des licences avec des demandeurs du monde entier, soit sans frais soit à des termes et conditions raisonnables et non discriminatoires. À ce propos, la déclaration du détenteur des droits de propriété est enregistrée à la CEI.

Des informations peuvent être demandées à:

[AB]	ABB AG Heidelberg Allemagne
[FE]	Festo AG Esslingen Allemagne
[SI]	Siemens AG Otto-Hahn-Ring 6 81739 Munich Allemagne
[SK]	Sick AG Waldkirch Allemagne

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

L'attention est d'autre part attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux qui ont été mentionnés ci-dessus. La CEI ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

L'ISO (www.iso.org/patents) et la CEI (http://patents.iec.ch) maintiennent des bases de données, consultables en ligne, des droits de propriété pertinents à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir l'information la plus récente concernant les droits de propriété.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

AUTOMATES PROGRAMMABLES –

Partie 9: Interface de communication numérique point à point pour petits capteurs et actionneurs (SDCI)

1 Domaine d'application

La présente partie de la CEI 61131 spécifie une technologie d'interface de communication numérique point à point pour petits capteurs et actionneurs – SDCI – (généralement connue sous l'appellation IO-Link™[2]); elle permet d'étendre les interfaces classiques d'entrées et de sorties numériques (telles que définies dans la CEI 61131-2) en une liaison de communication point à point. Cette technologie permet de transférer des paramètres aux Dispositifs et de fournir au système d'automatisation des informations de diagnostic depuis les Dispositifs.

Cette technologie est principalement destinée à l'utilisation en automatisation d'usine avec de simples capteurs et actionneurs intégrant de petits microcontrôleurs économiques.

La présente partie spécifie les services et le protocole de communication SDCI (couche physique, couche de liaison de données et couche Application conformément au modèle de référence ISO/OSI) pour les Maîtres et Dispositifs SDCI.

La présente partie inclut également des exigences d'essai CEM.

La présente partie ne couvre pas les interfaces ou systèmes de communication comportant des connexions multipoints ou à liaisons multiples, ou encore l'intégration de SDCI dans des systèmes de niveau supérieur tels que les bus de terrain.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI 60947-5-2, *Appareillage à basse tension – Partie 5-2: Appareils et éléments de commutation pour circuits de commande – Détecteurs de proximité*

CEI 61000-4-2, *Compatibilité électromagnétique (CEM) – Partie 4-2: Techniques d'essai et de mesure – Essai d'immunité aux décharges électrostatiques*

CEI 61000-4-3, *Compatibilité électromagnétique (CEM) – Partie 4-3: Techniques d'essai et de mesure – Essai d'immunité aux champs électromagnétiques rayonnés aux fréquences radioélectriques*

CEI 61000-4-4, *Compatibilité électromagnétique (CEM) – Partie 4-4: Techniques d'essai et de mesure – Essai d'immunité aux transitoires électriques rapides en salves*

---

[2] IO-Link™ est une marque commerciale de "IO-Link Consortium". Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme ne nécessite pas l'utilisation des logotypes déposés pour IO-Link™. L'utilisation des logotypes déposés pour IO-Link™ nécessite l'autorisation du "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

CEI 61000-4-5, *Compatibilité électromagnétique (CEM) – Partie 4-5: Techniques d'essai et de mesure – Essai d'immunité aux ondes de choc*

CEI 61000-4-6, *Compatibilité électromagnétique (CEM) – Partie 4-6: Techniques d'essai et de mesure – Immunité aux perturbations conduites, induites par les champs radioélectriques*

CEI 61000-4-11, *Compatibilité électromagnétique (CEM) – Partie 4-11: Techniques d'essai et de mesure – Essais d'immunité aux creux de tension, coupures brèves et variations de tension*

CEI 61000-6-2, *Compatibilité électromagnétique (CEM) – Partie 6-2: Normes génériques – Immunité pour les environnements industriels*

CEI 61000-6-4, *Compatibilité électromagnétique (CEM) – Partie 6-4: Normes génériques – Norme sur l'émission pour les environnements industriels*

CEI 61076-2-101, *Connecteurs pour équipements électroniques – Exigences de produit – Partie 2-101: Connecteurs circulaires – Spécification particulière pour les connecteurs M12 à vis*

CEI 61131-1, *Automates programmables – Partie 1: Informations générales*

CEI 61131-2, *Automates programmables – Partie 2: Exigences et essais des équipements*

CEI/TR 62390, *Common automation device – Profile guideline* (disponible en anglais seulement)

ISO/CEI 646:1991, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

ISO/CEI 2022, *Technologies de l'information – Structure de code de caractères et techniques d'extension* (disponible en anglais seulement)

ISO/CEI 10646, *Technologies de l'information — Jeu universel de caractères codés sur plusieurs octets (JUC)* (disponible en anglais seulement)

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Conventions pour la définition des services OSI*

ISO/CEI 19505 (toutes les parties), *Technologies de l'information — Langage de modélisation unifié OMG (OMG UML)* (disponible en anglais seulement)

ISO 1177, *Traitement de l'information – Structure des caractères pour la transmission arythmique et synchrone orientée caractère*

IEEE Std 754-2008, *IEEE Standard for Binary Floating-Point Arithmetic* (disponible en anglais seulement)

Internet Engineering Task Force (IETF): RFC 5905 – *Network Time Protocol Version 4: Protocol and Algorithms Specification;* consultable à l'adresse < www.ietf.org >

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions de la CEI 61131-1 et de la CEI 61131-2, ainsi que les suivants s'appliquent.

3.1.1
adresse
partie d'une commande de séquence M utilisée pour référencer des données dans des catégories de données d'un canal de communication

3.1.2
couche d'application
AL
<SDCI> partie du protocole chargée de la transmission des objets de Données de Processus et des objets de Données à la Demande

3.1.3
paramètre de bloc
accès cohérent à des paramètres via de multiples Index ou Sous-index

3.1.4
somme de contrôle
<SDCI> partie complémentaire des mesures globales d'intégrité des données dans la couche de liaison de données en supplément au bit de parité de l'UART

3.1.5
CHKPDU
données de protection de l'intégrité dans un canal de communication ISDU générées par traitement XOR des octets d'une demande ou d'une réponse

3.1.6
commutation codée
communication SDCI, fondée sur les niveaux de signaux binaires normalisés de la CEI 61131-2

3.1.7
COM1
mode de communication SDCI à une vitesse de transmission de 4,8 kbit/s

3.1.8
COM2
mode de communication SDCI à une vitesse de transmission de 38,4 kbit/s

3.1.9
COM3
mode de communication SDCI à une vitesse de transmission de 230,4 kbit/s

3.1.10
COMx
un des trois modes de communication SDCI possibles: COM1, COM2 ou COM3

3.1.11
canal de communication
connexion logique entre Maître et Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Note 1 à l'article: Il est défini quatre canaux de communication: un canal de processus, un canal de page et d'ISDU (pour les paramètres), et un canal de diagnostic.

3.1.12
erreur de communication
perturbation imprévue du protocole de transmission SDCI

3.1.13
durée de cycle
temps nécessaire à la transmission d'une séquence M entre un Maître et son Dispositif, y compris le temps de repos qui suit

3.1.14
Dispositif
homologue passif unique d'un Maître, tel qu'un capteur ou un actionneur

Note 1 à l'article: "Dispositif" avec un D majuscule désigne un équipement SDCI, tandis que "dispositif" avec un d minuscule est une désignation générique.

3.1.15
Paramètres Directs
paramètres à adressage direct (page) transférés de manière acyclique via le canal de communication de page sans acquittement

3.1.16
paramètre dynamique
partie d'un jeu de paramètres de Dispositif définie par des interfaces utilisateur embarquées, telles que des boutons d'apprentissage ou des panneaux de commande, et qui s'ajoutent aux paramètres statiques

3.1.17
Evénement
instance d'un changement de conditions dans un Dispositif

Note 1 à l'article: "Événement" avec un É majuscule désigne un Événement SDCI, tandis que "événement" avec un é minuscule est utilisé de manière générique.

Note 2 à l'article: Un Événement est indiqué par le Drapeau (indicateur) d'Événement dans les informations cycliques d'état du Dispositif; le transfert acyclique des données d'Événement (en général des informations de diagnostic) est réalisé par l'intermédiaire du canal de communication de diagnostic.

3.1.18
Repli
passage d'un port de la commutation codée au mode signal de commutation

3.1.19
niveau de vérification
degré de vérification de l'identité du Dispositif

3.1.20
entrelacement
échange de données cycliques segmentées pour des Données de Processus comportant plus de 2 octets au cours de cycles consécutifs

3.1.21
ISDU
unité de données de service indexée utilisée pour la transmission acyclique avec acquittement de paramètres qui peuvent être segmentés en un certain nombre de séquences M

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.22
héritage Dispositif ou Maître
Dispositif ou Maître conçu conformément à [8][3]

3.1.23
Séquence M
séquence de deux messages comprenant un message du Maître et le message du Dispositif résultant

3.1.24
commande de Séquence M
premier octet dans un message du Maître indiquant l'opération de lecture/écriture, le type de canal de communication et l'adresse, par exemple décalage ou régulation de flux

3.1.25
erreur de Séquence M
contenu de message imprévu ou erroné, ou absence de réponse

3.1.26
type de Séquence M
format particulier de séquence M parmi un ensemble spécifié de formats de séquence M

3.1.27
Maître
homologue actif connecté par l'intermédiaire de ports à un jusqu'à *n* Dispositifs et qui fournit une interface avec la passerelle vers les systèmes de communication de niveau supérieur ou des automates programmables

Note 1 à l'article: "Maître" avec un M majuscule désigne un équipement SDCI, tandis que "maître" avec un m minuscule est une désignation générique.

3.1.28
message
<SDCI> séquence de trames UART transférées d'un Maître à son Dispositif ou réciproquement selon les règles du protocole SDCI

3.1.29
Données à la Demande
données transmises de manière acyclique à la demande de l'application du Maître et constituées de paramètres ou de données d'Événement

3.1.30
couche physique
première couche du modèle de référence ISO-OSI, qui fournit les moyens mécaniques, électriques, fonctionnels et procéduraux pour activer, maintenir et désactiver des connexions physiques de transmission binaire entre entités de liaison de données

Note 1 à l'article: La couche physique fournit également des moyens de réactivation et procédures de repli.

[SOURCE: ISO/CEI 7498-1:1994, 7.7.2, modifiée – texte extrait du paragraphe, note ajoutée]

3.1.31
port
interface de support de communication du Maître vers un Dispositif

---

3 Les nombres entre crochets se rapportent à la Bibliographie.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.1.32
mode de fonctionnement du port
état d'un port de Maître qui peut être:INACTIVE, DO, DI, FIXEDMODE ou SCANMODE

3.1.33
Données de Processus
valeurs d'entrée ou de sortie, à partir de ou vers un procédé d'automatisation en continu ou par lot transférées de manière cyclique, hautement prioritaire et selon une planification configurée automatiquement après démarrage d'un Maître

3.1.34
cycle de Données de Processus
transfert complet de toutes les Données de Processus à partir de ou vers un Dispositif individuel qui peut inclure plusieurs cycles en cas de segmentation (entrelacement)

3.1.35
paramètre unique
accès de paramètre indépendant via un Index ou Sous-index unique

3.1.36
SIO
mode de fonctionnement de port conforme aux entrées et sorties numériques définies dans la CEI 61131-2 qui est établi après une mise sous tension, un repli ou un échec de tentatives de communication

3.1.37
paramètre statique
partie d'un jeu de paramètres de Dispositif à sauvegarder dans un Maître dans le cas d'un remplacement sans outillage technique

3.1.38
signal de commutation
signal binaire en provenance de ou vers un Dispositif, en mode SIO (par opposition à la communication SDCI à "commutation codée")

3.1.39
gestion système
<SDCI> moyens de commande et de coordination des couches de communication internes et des exceptions dans le Maître et ses ports, ainsi que dans chaque Dispositif

3.1.40
trame UART
<SDCI> séquence binaire commençant par un bit de départ, suivi de huit bits transportant un octet de données, suivi par un bit de parité paire et se terminant par un bit d'arrêt

3.1.41
réactivation
procédure permettant de faire passer un Dispositif du mode SIO en SDCI

3.1.42
demande de réactivation
service de couche physique utilisé par le Maître pour lancer la réactivation d'un Dispositif, et le mettre en un état prêt à recevoir

3.2 Symboles et abréviations

Δf_{DTRM}	écart admissible par rapport au débit de transfert de données (mesuré en %)
ΔVS	ondulation de l'alimentation (mesurée en V)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

AL	couche d'application (application layer)
BEP	probabilité d'erreur sur les bits (bit error probability)
C/Q	connexion pour signal (SIO) de communication (C) ou de commutation (Q)
CL_{eff}	capacité effective totale des câbles (mesurée en nF)
CQ	capacité d'entrée au niveau de la connexion C/Q (mesurée en nF)
DI	entrée numérique
DL	couche de liaison de données (data link layer)
DO	sortie numérique (digital output)
f_{DTR}	débit de transfert de données (mesuré en bit/s)
H/L	signal haut/bas à la sortie du récepteur
E/S	entrée/sortie
ILL	courant de charge d'entrée au niveau de C/Q d'entrée vers $V0$ (mesuré en A)
IODD	descriptif de dispositif d'E/S (IO device description) (voir 10.8)
IQ	courant d'attaque à l'état de fonctionnement saturé ON (mesuré en A)
IQH	courant d'attaque du côté haut, à l'état de fonctionnement saturé ON (mesuré en A)
IQL	courant d'attaque du côté bas, à l'état de fonctionnement saturé ON (mesuré en A)
$IQPK$	courant d'attaque maximal, à l'état de fonctionnement non saturé ON (mesuré en A)
$IQPKH$	courant d'attaque maximal du côté haut, à l'état de fonctionnement non saturé ON (mesuré en A)
$IQPKL$	courant d'attaque maximal du côté bas, à l'état de fonctionnement non saturé ON (mesuré en A)
IQQ	courant passif au niveau de C/Q d'entrée vers $V0$ avec pilotes de sortie inactifs (mesuré en A)
IQ_{WU}	amplitude du courant de demande de réactivation du Maître, mesurée en A
IS	courant d'alimentation au niveau de $V+$ (mesuré en A)
$ISIR$	capacité de fourniture d'impulsion de courant au niveau de $V+$ (mesurée en A)
DEL	diode électroluminescente
L-	alimentation (-)
L+	alimentation (+)
N24	alimentation 24 V supplémentaire (-)
n_{WU}	nombre de nouvelles tentatives de réactivation
On/Off	signal de commutation Actif/Non actif du circuit d'attaque
OD	données à la demande (On-request Data)
OVD	détection de surcharge du signal (signal overload detect)
P24	alimentation 24 V supplémentaire (+)
PD	données de processus (process data)
PDCT	outil de configuration des ports et Dispositifs (port and device configuration tool)
PL	couche physique (physical layer)
PLC	automate programmable (programmable logic controller)
PS	alimentation (mesurée en V)
r	temps nécessaire pour atteindre un niveau stable en référence au début du bit de départ (mesuré en T_{BIT})
RL_{eff}	résistance de boucle de câble, mesurée en Ω
s	temps nécessaire pour sortir d'un niveau stable en référence au début du bit de départ (mesuré en T_{BIT})
SDCI	interface de communication numérique point à point (single-drop digital communication interface)
SIO	entrée/sortie normalisée (standard input output) (mode de commutation numérique) [CEI 61131-2]
SM	gestion système (system management)
t_1	délai de transfert de trame UART sur le Maître (mesuré en T_{BIT})
t_2	délai de transfert de trame UART sur le Dispositif (mesuré en T_{BIT})

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

t_A	délai de réponse sur le Dispositif (mesuré en T_{BIT})
T_{BIT}	temps binaire (mesuré en s)
t_{CYC}	durée de cycle sur un niveau de séquence M (mesurée en s)
t_{DF}	temps de décroissance (mesuré en s)
T_{DMT}	temps de retard pendant l'établissement de la communication du port du Maître (mesuré en T_{BIT})
T_{DR}	temps de croissance (mesuré en s)
T_{DSIO}	temps de retard sur le dispositif pour le passage en mode SIO suite à une demande de réactivation (mesuré en s)
T_{DWU}	délai de nouvelle tentative de réactivation (mesurée en s)
$t_{séquence\ M}$	durée de séquence M (mesurée en T_{BIT})
t_{idle}	temps au repos entre deux séquences M (mesuré en s)
t_H	temps de détection d'un niveau haut (mesuré en s)
t_L	temps de détection d'un niveau bas (mesuré en s)
t_{ND}	temps de suppression du bruit (mesuré en s)
T_{RDL}	délai de l'état prêt à la réactivation après mise sous tension (mesuré en s)
T_{REN}	délai d'activation de la réception (mesuré en s)
T_{SD}	temps de détection du dispositif (mesuré en s)
T_{WU}	durée d'impulsion de demande de réactivation (mesurée en s)
UART	émetteur-récepteur asynchrone universel (universal asynchronous receiver transmitter)
UML	langage de modélisation unifié (Unified Modelling Language) [ISO/CEI 19505]
$V+$	tension en L+
$V0$	tension en L-
$VD+_L$	chute de tension sur la ligne entre les connexions L+ sur le Maître et le Dispositif (mesurée en V)
$VD0_L$	chute de tension sur la ligne entre les connexions L- sur le Maître et le Dispositif (mesurée en V)
VDQ_L	chute de tension sur la ligne entre les connexions C/Q sur le Maître et le Dispositif (mesurée en V)
$VHYS$	hystérésis de la tension de seuil du récepteur (mesurée en V)
VI	tension d'entrée au niveau de la connexion C/Q en référence à $V0$ (mesurée en V)
VIH	plage de tension d'entrée au niveau de la connexion C/Q pour un signal haut (mesurée en V)
VIL	plage de tension d'entrée au niveau de la connexion C/Q pour un signal bas (mesurée en V)
VRQ	tension résiduelle sur le circuit d'attaque à l'état de fonctionnement saturé ON (mesurée en V)
$VRQH$	tension résiduelle sur le circuit d'attaque du côté haut, à l'état de fonctionnement ON (mesurée en V)
$VRQL$	tension résiduelle sur le circuit d'attaque du côté bas, à l'état de fonctionnement saturé ON (mesurée en V)
VTH	tension de seuil du récepteur en référence à $V0$ (mesurée en V)
$VTHH$	tension de seuil du récepteur pour une détection sûre d'un signal haut (mesurée en V)
$VTHL$	tension de seuil du récepteur pour une détection sûre d'un signal bas (mesurée en V)
WURQ	impulsion de demande de réactivation (wake-up request pulse)

3.3 Conventions

3.3.1 Généralités

Le modèle de service, les primitives de service et les diagrammes présentés dans la présente norme sont des descriptions entièrement abstraites. La mise en œuvre des services peut refléter des sujets individuels et peut différer en fonction de la situation.

3.3.2 Paramètres de services

Des primitives de service sont utilisées pour représenter des interactions entre fournisseur /consommateur de services (ISO/CEI 10731). Elles acheminent des paramètres qui

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

définissent les informations disponibles dans l'interaction fournisseur/ consommateur. Il n'est pas nécessaire de déclarer explicitement chacun des paramètres d'une quelconque interface particulière.

La spécification des services dans la présente norme utilise un format tabulaire pour décrire les paramètres constitutifs des primitives de service. Les paramètres applicables à chaque groupe de primitives de service sont présentés dans des tableaux. Chaque tableau est constitué d'au maximum cinq colonnes:

1) nom du paramètre;
2) primitive de demande (.req);
3) primitive d'indication (.ind);
4) primitive de réponse (.rsp); et
5) primitive de confirmation (.cnf).

Un paramètre (ou élément constitutif de ce paramètre) est présenté dans chaque rangée de chaque tableau. Dans les colonnes des primitives de service concernées, un code permet de spécifier le type d'utilisation du paramètre sur la primitive spécifiée dans la colonne:

M Le paramètre est obligatoire pour la primitive.

U Le paramètre est une option utilisateur; sa mise en œuvre dépend de son utilisation dynamique par l'utilisateur du service. S'il n'est pas fourni, on suppose une valeur par défaut du paramètre.

C Le paramètre est conditionné par d'autres paramètres ou par l'environnement de l'utilisateur du service.

– Le paramètre n'est jamais présent.

S Le paramètre est un élément sélectionné.

Certaines entrées sont en outre qualifiées par un symbole entre parenthèses. Il peut s'agir:

a) d'une contrainte spécifique au paramètre; "(=)" indique que le paramètre est sémantiquement équivalent au paramètre dans la primitive de service qui se trouve immédiatement à sa gauche dans le tableau;
b) d'une indication selon laquelle une éventuelle note s'applique à l'entrée "(n)" et que la note "n" suivante contient des informations supplémentaires concernant le paramètre et son utilisation.

3.3.3 Procédures de services

Les procédures sont définies en termes:

- d'interactions entre entités d'application par échange d'unités de données de protocole, et
- d'interactions entre un fournisseur de services de couche de communication et un consommateur de services de couche de communication dans le même système par invocation de primitives de service.

Ces procédures s'appliquent à des instances de communication entre systèmes qui prennent en charge des services de communication comportant des contraintes de temps au sein des couches de communication.

3.3.4 Attributs de services

La nature des différents services (Maître et Dispositif) est caractérisée par des attributs. Tous les services sont définis du point de vue de la couche concernée vers la couche au-dessus.

I Initiateur d'un service (vers la couche au-dessus)

R Destinataire (répondeur) d'un service (à partir de la couche au-dessus)

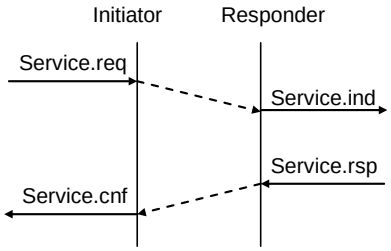
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

3.3.5 Figures

Les conventions suivantes sont utilisées dans les figures illustrant la structure et les services de couches de protocole:

- une flèche avec seulement un nom de service représente à la fois une demande et la confirmation correspondante, la demande étant effectuée dans le sens de la flèche;
- une demande sans confirmation, ainsi que toutes les indications et réponses sont libellées en tant que telles (c'est-à-dire service.req, service.ind, service.rsp).

La Figure 1 illustre un exemple de service confirmé.



Légende

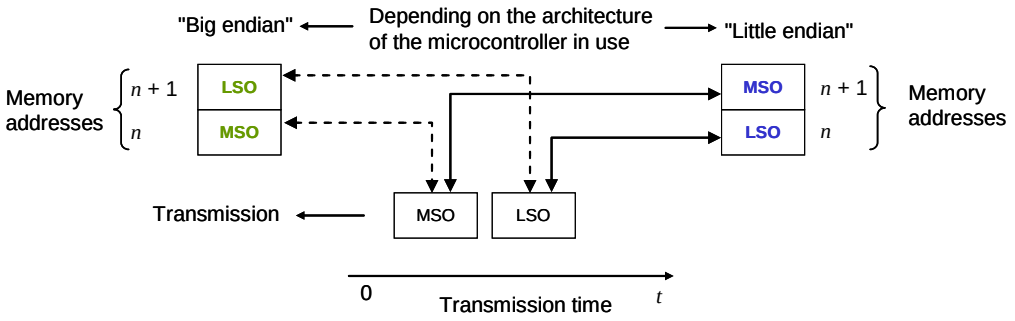
Anglais	Français
Initiator	Initiateur
Responder	Répondeur (destinataire)

Figure 1 – Exemple de service confirmé

3.3.6 Ordre de transmission des octets

La Figure 2 illustre la manière dont des types de données à base de MOTS sont transférés de la mémoire au support de transmission et réciproquement (c'est-à-dire l'octet de poids fort transmis en premier, voir 7.3.3.2 et 7.3.6.1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

MSO = octet de poids fort
LSO = octet de poids faible

Légende

Anglais	Français
Transmission time	Temps de transmission
Transmission	Transmission
"Big endian"	"Gros-boutiste"
"Little endian"	"Petit-boutiste"
Memory addresses	Adresses mémoire
Depending on the architecture of the microcontroller in use	En fonction de l'architecture du microcontrôleur utilisé

Figure 2 – Ordre de stockage en mémoire et de transmission pour des types de données à base de MOTS

3.3.7 Descriptifs comportementaux

Les descriptifs comportementaux utilisent les notations du langage UML 2 (ISO/CEI 19505) (par exemple des diagrammes d'états, de séquences, d'activités, des chronogrammes, des conditions de protection). Les tableaux de transitions d'états correspondants sont présentés selon la CEI/TR 62390.

Du fait des restrictions liées à l'outil de conception, les exceptions suivantes s'appliquent. Pour les diagrammes d'états, un paramètre de service (en lettres majuscules) est relié au nom de service par un caractère de soulignement, comme par exemple dans DL_SetMode_INACTIVE. Pour les diagrammes de séquences, la primitive de service est reliée par un caractère de soulignement au lieu d'un point, et le paramètre de service est ajouté entre parenthèses, comme par exemple dans DL_Event_ind (OPERATE). Les contraintes temporelles sont libellées "tm(temps en ms)".

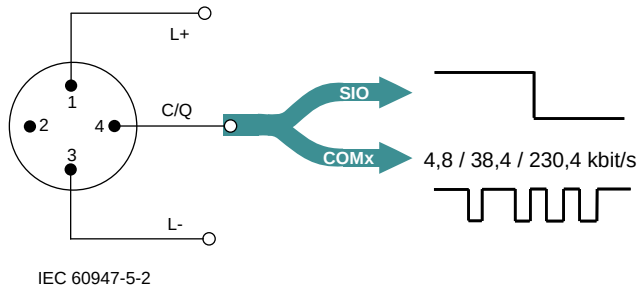
Les invocations de services reçues en asynchrone ne sont pas modélisées en détail dans les diagrammes d'états.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

4 Aperçu global de l'interface SDCI (IO-Link™[4])

4.1 Objectif de la technologie

La Figure 3 illustre le concept de base de l'interface SDCI.



Pin	Signal	Definition	Standard
1	L+	24 V	IEC 61131-2
2	I/Q	Not connected, DI, or DO	IEC 61131-2
3	L-	0 V	IEC 61131-2
4	Q	"Switching signal" DI, DO (SIO)	IEC 61131-2
	C	"Coded switching" (COM1, COM2, COM3)	IEC 61131-9

Légende

Anglais	Français
IEC	CEI
Pin	Broche
Signal	Signal
Definition	Définition
Standard	Norme
"Switching Signal"	"Signal de commutation"
"Coded switching"	"Commutation codée"
Not connected, DI, or DO	Non connecté, DI ou DO

Figure 3 – Compatibilité de la SDCI avec la CEI 61131-2

La technologie d'interface de communication numérique point à point pour petits capteurs et actionneurs – SDCI – (généralement connue sous l'appellation IO-Link™) définit une voie de migration, à partir des interfaces existantes d'entrées et de sorties numériques pour des Dispositifs à commutation 24 V telles que définies dans la CEI 61131-2, permettant d'obtenir une liaison de communication point à point. Ainsi, les modules d'E/S numériques des périphériques de bus de terrain existants, par exemple, peuvent être remplacés par des modules Maître SDCI fournissant à la fois des interfaces DI/DO classiques et l'interface SDCI. La technologie de transmission analogique peut être remplacée par la technologie SDCI qui associe sa robustesse ainsi que ses fonctionnalités de paramétrage et de diagnostic à l'économie d'efforts de conversion numérique/analogique et analogique/numérique.

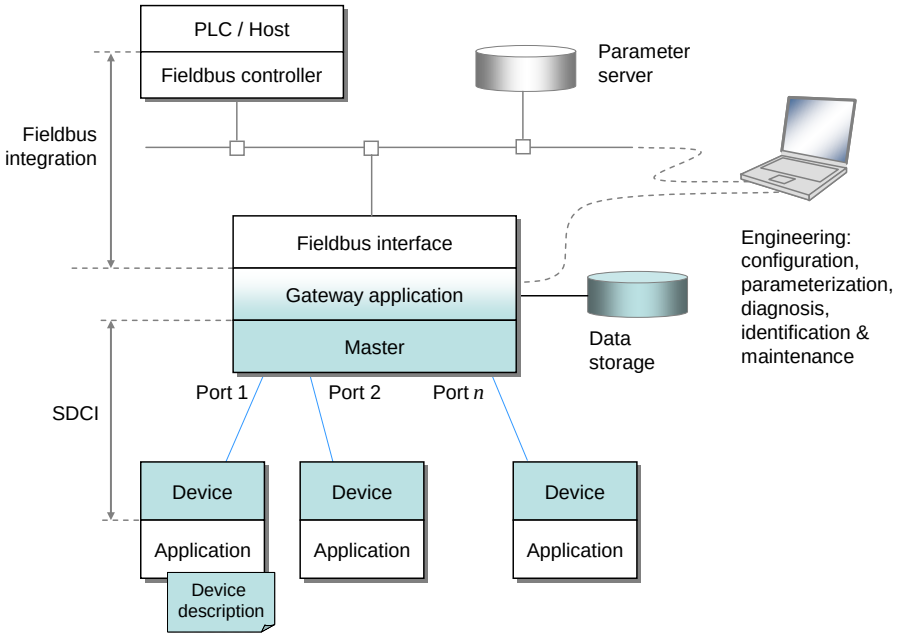
4.2 Positionnement dans la hiérarchie de l'automatisation

La Figure 4 présente le domaine de la technologie SDCI au sein de la hiérarchie de l'automatisation.

---

[4] IO-Link™ est une marque commerciale de "IO-Link Consortium". Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à la présente norme ne nécessite pas l'utilisation des logotypes déposés pour IO-Link™. L'utilisation des logotypes déposés pour IO-Link™ nécessite l'autorisation du "IO-Link Consortium".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device	Dispositif
Application	Application
Fieldbus interface	Interface de bus de terrain
Data storage	Stockage des données
Parameter server	Serveur de paramètres
PLC / Host	Automate programmable / hôte
Fieldbus integration	Intégration au bus de terrain
Device description	Description de Dispositif
Engineering: configuration, parameterization, diagnosis, identification & maintenance	Personnel technique d'ingénierie: configuration, paramétrage, diagnostic, identification & maintenance
Fieldbus controller	Contrôleur de bus de terrain
Gateway application	Application passerelle
Master	Maître
Port	Port

Figure 4 – Domaine de la technologie SDCI au sein de la hiérarchie de l'automatisation

La technologie SDCI définit une interface générique pour la connexion de capteurs et d'actionneurs à une unité Maître, qui peut être associée à des fonctionnalités de passerelle pour constituer un nœud déporté d'E/S de bus de terrain.

Le point de départ de la conception de l'interface SDCI est l'entrée numérique (DI) 24 V conventionnelle définie dans la CEI 61131-2 et l'interface de sortie (DO) spécifiée dans le Tableau 6. L'interface SDCI offre ainsi la connectivité de capteurs 24 V conventionnels ("signaux de commutation") comme mode de fonctionnement par défaut. Une connectivité supplémentaire est également fournie pour des actionneurs si un port est configuré en "mode de communication point à point".

De nos jours, de nombreux capteurs et actionneurs sont déjà équipés de microcontrôleurs proposant une interface UART qui peut être étendue par ajout de quelques composants matériels et d'un logiciel protocolaire prenant en charge la communication SDCI. Ce second

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

mode de fonctionnement utilise la "commutation codée" de ligne de signal d'E/S 24 V. Une fois activé, le mode SDCI prend en charge le paramétrage, l'échange des données cycliques, les rapports de diagnostic, les informations d'identification & de maintenance, ainsi que le stockage externe des paramètres pour la sauvegarde des réglages des Dispositifs et un rechargement rapide des dispositifs de remplacement. Les capteurs et actionneurs ayant une capacité SDCI sont appelés "Dispositifs" dans la présente norme. Pour améliorer les performances au démarrage, ces Dispositifs comportent généralement une mémoire non volatile de stockage des paramètres.

NOTE La configuration et le paramétrage des Dispositifs sont pris en charge par un descriptif de dispositifs fondé sur le langage XML (voir [6]), non couvert dans la présente norme.

4.3 Câblage, connecteurs et alimentation

La connexion par défaut (classe de port A) comporte 4 broches (voir la Figure 3). Le câblage par défaut pour la classe de port A est conforme à la CEI 60947-5-2 et n'utilise que trois conducteurs: 24 V, 0 V et une ligne de signal. Le quatrième conducteur peut être utilisé comme ligne de signal supplémentaire conformément à la CEI 61131-2.

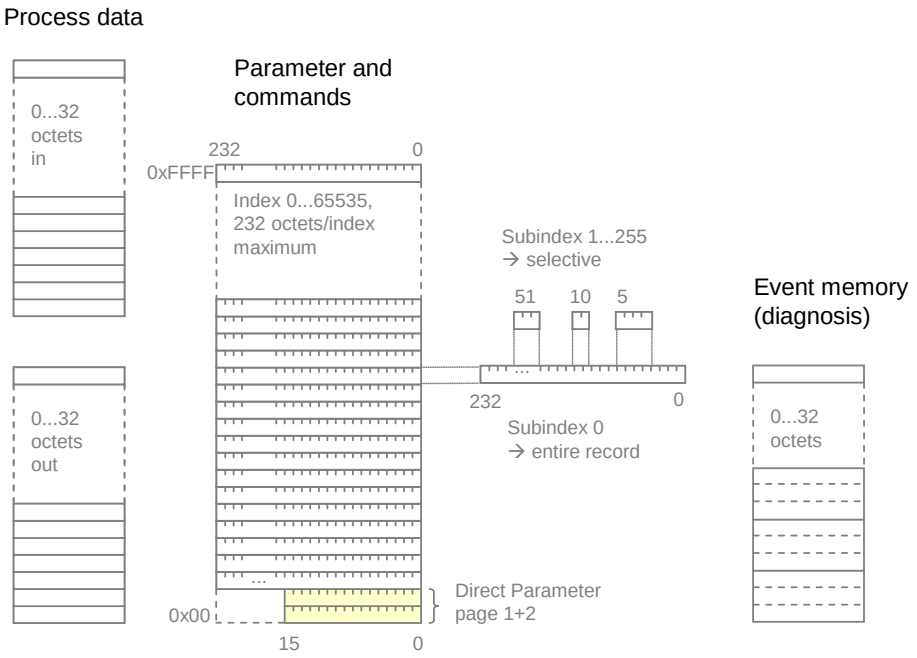
Des connexions à cinq broches (classe de port B) sont spécifiées pour des Dispositifs nécessitant une puissance supplémentaire fournie à partir d'une alimentation 24 V indépendante.

NOTE Un Dispositif de classe de port A utilisant le quatrième conducteur est incompatible avec un Maître de classe de port B.

La longueur maximale des câbles est de 20 m; il n'est pas exigé de blindage.

4.4 Fonctionnalités de communication de l'interface SDCI

Le modèle de Dispositif générique est illustré en Figure 5 et expliqué dans les alinéas suivants.



Légende

Anglais	Français
Process data	Données de processus
0…32 octets out	0 à 32 octets de sortie

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
0...32 octets in	0 à 32 octets d'entrée
Index 0...65535,	Index 0 à 65535,
232 octets/index maximum	232 octets/index au maximum
Subindex 0	Sous-index 0
entire record	ensemble de l'enregistrement
Parameter and commands	Paramètres et commandes
0...32 octets	0 à 32 octets
Event memory (diagnosis)	Mémoire d'Événements (diagnostic)
Subindex 1...255	Sous-index 1 à 255
selective	sélectif
Direct Parameter page 1+2	Pages 1+2 de Paramètres Directs

Figure 5 – Modèle de Dispositif générique pour SDCI (vue du côté Maître)

Un Dispositif peut recevoir des Données de Processus (de sortie) pour commander un processus d'automatisation par lot ou en continu ou envoyer des Données de Processus (d'entrée) représentant son état courant ou des valeurs mesurées. Le Dispositif fournit généralement des paramètres permettant à l'utilisateur de configurer ses fonctions et répondre ainsi à des besoins particuliers. Pour tenir compte de telles circonstances, il est défini un vaste espace de paramétrage auquel il est possible d'accéder par l'intermédiaire d'un Index (0 à 65 535; à structure prédéfinie) et d'un Sous-index (0 à 255).

Les deux premières entrées de l'index, 0 et 1, sont réservées aux pages 1 et 2 des Paramètres Directs, d'au maximum 16 octets chacune. La page de paramètres 1 est principalement dédiée aux commandes du Maître telles que démarrage et repli du Dispositif, récupération d'informations de fonctionnement et d'identification spécifiques au Dispositif. La page de paramètres 2 prévoit un maximum de 16 octets de paramètres spécifiques au Dispositif.

Chacun des autres index (2 à 65 535) permet d'accéder à un enregistrement d'une taille maximale de 232 octets. Le sous-index 0 spécifie la transmission de l'enregistrement complet indiqué par l'Index et les autres sous-index spécifient le transfert d'éléments de données sélectionnés dans l'enregistrement.

Dans un enregistrement, les éléments de données individuels peuvent commencer à partir de n'importe quel décalage binaire, et leur longueur peut aller de 1 bit à 232 octets; cependant, le nombre total d'éléments de données dans l'enregistrement ne peut dépasser 255. L'organisation des éléments de données dans un enregistrement est spécifiée dans le Descriptif de Dispositif d'E/S (IODD).

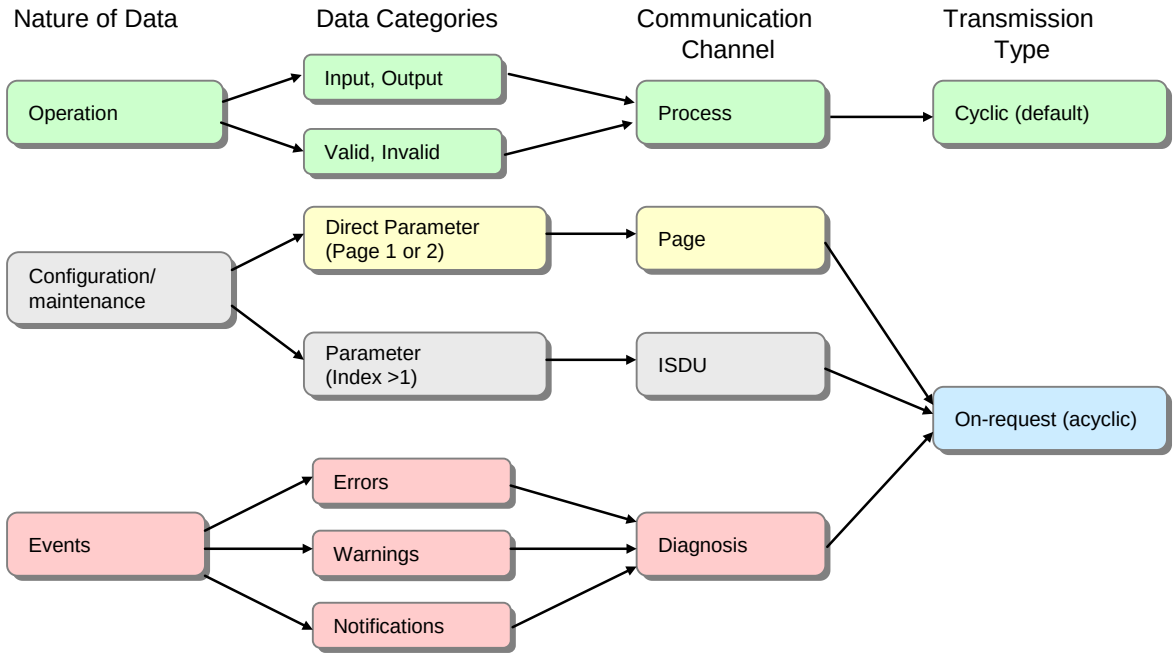
Toute modification d'une condition de Dispositif qui nécessite un rapport ou une intervention est stockée dans une mémoire d'Événements avant transmission. Un drapeau d'Événement est ensuite établi dans l'échange des données cycliques pour indiquer l'existence d'un Événement.

La communication entre un Maître et un Dispositif est effectuée en point à point et se fonde sur le principe de l'envoi d'un message de demande par un Maître puis l'envoi d'un message de réponse par un Dispositif (voir la Figure 36). Pris ensemble, ces messages sont appelés une Séquence M. Plusieurs types de Séquence M sont définis pour prendre en charge les exigences de l'utilisateur en termes de transmission de données (voir la Figure 37).

Comme illustré en Figure 6, diverses catégories de données sont transmises par des canaux de communication individuels de la couche de liaison de données.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- Les données opérationnelles telles que les entrées et sorties de Dispositif sont transmises de manière cyclique par l'intermédiaire d'un canal de processus. Les données opérationnelles peuvent également être associées à des qualificatifs tels que valide/non valide.
- Les paramètres de configuration et de maintenance sont transmis de manière acyclique. Un canal de page est prévu pour accéder directement aux pages de paramètres 1 et 2, et un canal ISDU est utilisé pour accéder à des paramètres et commandes supplémentaires.
- Les événements de Dispositifs sont transmis de manière acyclique par l'intermédiaire d'un canal de diagnostic. Les rapports d'événements de Dispositifs comportent trois niveaux de sévérité: erreur, avertissement et notification.



Légende

Anglais	Français
Nature of Data	Nature des données
Data Categories	Catégories de données
Communication Channel	Canal de communication
Transmission Type	Type de transmission
Operation	Fonctionnement
Input, Output	Entrée, Sortie
Valid, Invalid	Valide, Non valide
Process	Processus
Cyclic (default)	Cyclique (par défaut)
Configuration/maintenance	Configuration/maintenance
Direct Parameter (Page 1 or 2)	Paramètres Directs (Page 1 ou 2)
Page	Page
Parameter (Index >1)	Paramètre (Index >1)
On-request (acyclic)	À la demande (acyclique)
Events	Événements
Errors	Erreurs
Warnings	Avertissements
Notifications	Notifications
Diagnosis	Diagnostic

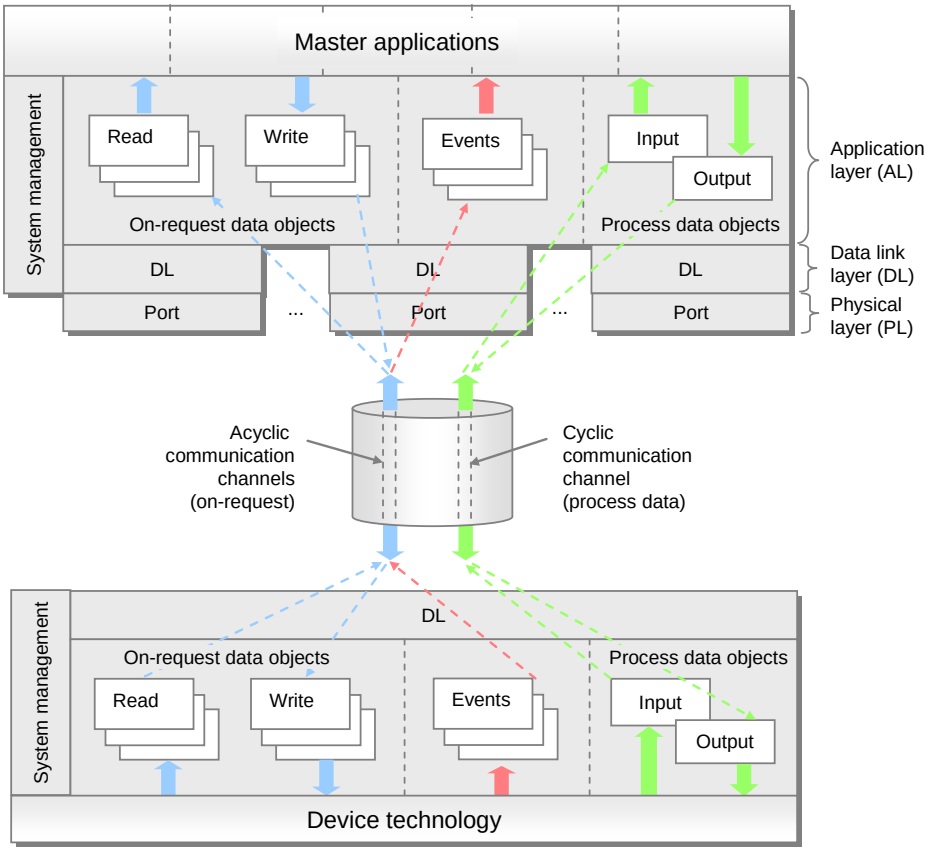
Figure 6 – Relation entre nature des données et types de transmission

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le premier octet d'un message de Maître commande le sens de transfert des données (lecture/écriture) et le type de canal de communication.

La Figure 7 montre que chaque port d'un Maître dispose de sa propre couche de liaison de données en interface avec une couche commune d'application du Maître. Dans la couche d'application, les services de la couche de liaison de données sont traduits en actions sur les objets de Données de Processus (entrée/sortie), les objets de Données à la Demande (lecture/écriture), et les événements. Les applications du Maître comprennent le Gestionnaire de Configuration (CM – Configuration Manager), le mécanisme de Stockage des Données (DS – Data Storage), l'Unité de Diagnostic (DU – Diagnosis Unit), l'Échange de Données à la Demande (ODE – On-request Data Exchange) et l'Échange de Données de Processus (PDE – Process Data Exchange).

La Gestion Système vérifie l'identification des Dispositifs connectés et règle les ports et les Dispositifs pour qu'ils correspondent à la configuration choisie et aux propriétés de ces mêmes Dispositifs connectés. Elle gère également, au démarrage par exemple, les diagrammes d'états des couches d'application (AL) et de liaison de données (DL).



Légende

Anglais	Français
Master applications	Applications du Maître
System management	Gestion Système
Read	Lecture
Write	Écriture
Events	Événements
Input	Entrée
Output	Sortie
Application layer (AL)	Couche d'application (AL)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
On-request data objects	Objets de Données à la Demande
Process data objects	Objets de Données de Processus
Data link layer (DL)	Couche de liaison de données (DL)
Physical layer (PL)	Couche physique (PL)
Acyclic communication channels (on-request)	Canaux de communication acyclique (à la demande)
Cyclic communication channel (process data)	Canal de communication cyclique (données de processus)
Device technology	Technologie du dispositif

Figure 7 – Transfert d'objets au niveau de la couche d'application (AL)

4.5 Rôle d'un Maître

Un Maître prend en charge 1 à *n* ports et les couches de liaison de données correspondantes. Au démarrage, il modifie les ports en fonction des modes sélectionnés par l'utilisateur: INACTIVE, DI, DO, FIXEDMODE ou SCANMODE. S'il y a demande de communication, le Maître utilise une impulsion de courant de réactivation spéciale pour lancer la communication avec le Dispositif. Ensuite, le Maître règle automatiquement la vitesse de transmission sur COM1, COM2 ou COM3 (voir le Tableau 8) et vérifie la "personnalité" du Dispositif connecté, c'est-à-dire son VendorID (identifiant de fournisseur), son DeviceID (identifiant de Dispositif), et ses propriétés de communication.

En cas de discordance entre les paramètres du Dispositif et le jeu de paramètres enregistré dans le Maître, les paramètres dans le Dispositif sont écrasés (voir 11.3) ou, en fonction de la configuration, les paramètres enregistrés dans le Maître sont mis à jour.

Il est également possible de démarrer un dispositif en mode DI, de passer en communication SDCI pour la configuration et le paramétrage et d'utiliser ensuite la commande de repli (voir 11.8.5) pour revenir en mode DI de fonctionnement normal.

La coordination des ports incombe également au Maître que l'utilisateur peut configurer par sélection du mode de fonctionnement du port. En mode "FreeRunning" (fonctionnement libre), chaque port définit son propre cycle sur la base des propriétés du Dispositif connecté. En mode "MessageSync" (synchronisation des messages), les messages envoyés sur les ports connectés démarrent simultanément ou selon un échelonnement défini. En mode "FixedValue" (valeur fixe), chaque port utilise une durée de cycle fixe définie par l'utilisateur (voir 11.2.2.2).

Le Maître est chargé d'assembler et de désassembler toutes les données échangées avec les Dispositifs (voir Article 11).

Le Maître fournit une zone Stockage des Données d'au moins 2 048 octets par Dispositif pour sauvegarde des données de Dispositifs (voir 11.3). Pour ses propres besoins opérationnels, le Maître peut combiner ces données de Dispositif avec toutes autres données pertinentes et les mettre à la disposition d'applications de niveau supérieur à des fins de sauvegarde du Maître ou de contrôle des recettes (voir 11.8.3).

4.6 Configuration SDCI

La prise en charge technique d'un Maître est généralement assurée par un Outil de Configuration des Ports et Dispositifs (PDCT). Le PDCT permet de configurer les propriétés des ports et celles des Dispositifs (voir les paramètres illustrés en Figure 5). Cet outil associe un interpréteur de Descriptifs de Dispositifs d'E/S (IODD) et un configurateur (voir 11.7). L'IODD fournit l'ensemble des propriétés nécessaires à l'établissement de la communication ainsi que les paramètres et limites correspondantes nécessaires à l'établissement de la

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

fonction souhaitée d'un capteur ou d'un actionneur. Le PDCT permet également de compiler les Données de Processus pour propagation sur le bus de terrain et inversement.

4.7 Mise en correspondance avec les bus de terrain

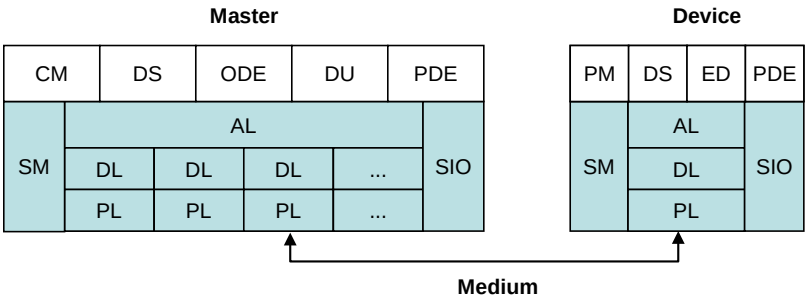
L'intégration d'un Maître au sein d'un système de bus de terrain, c'est-à-dire la définition des fonctions de passerelle d'échange de données avec des entités de niveau supérieur sur un bus de terrain, ne relève pas du domaine d'application de la présente norme.

EXEMPLE Ces fonctions comprennent la mise en correspondance des Données de Processus échangées, le paramétrage contrôlé par programme ou la mise en œuvre d'un serveur de paramètres distant, ou encore la propagation des informations de diagnostic.

L'intégration d'un PDCT dans des outils techniques d'ingénierie d'un bus de terrain particulier ne relève pas du domaine d'application de la présente norme.

4.8 Structure de la norme

La structure logique du Maître et du Dispositif est illustrée en Figure 8. L'Article 5 définit la Couche Physique (PL) de la SDCI, l'Article 6 décrit de manière détaillée le mode SIO. L'Article 7 définit les services et le protocole de la Couche de Liaison de Données (DL), la réactivation, les Séquences M et les modules de traitement de couche DL. L'Article 8 définit les services et le protocole de la Couche d'Application (AL) et l'Article 9 précise les responsabilités liées à la Gestion Système (SM).



Légende

Anglais	Français
Master	Maître
Device	Dispositif
Medium	Support

Figure 8 – Structure logique du Maître et du Dispositif

L'Article 10 spécifie les applications et fonctionnalités des Dispositifs, telles que l'Échange de Données de Processus (PDE), la Gestion des Paramètres (PM), le Stockage des Données (DS), et le Répartiteur d'Événements (ED). Les applications spécifiques à la technologie ne sont pas traitées dans la présente Norme. Elles peuvent être spécifiées dans des profils de familles de Dispositifs particulières.

L'Article 11 spécifie les applications et fonctionnalités du Maître, telles que l'Échange de Données de Processus (PDE), l'Échange de Données à la Demande (ODE), la Gestion de la Configuration (CM), le Stockage des Données (DS) et l'Unité de Diagnostic (DU).

Plusieurs annexes normatives et informatives sont incluses. L'Annexe A définit les types de Séquences M disponibles. L'Annexe B décrit les paramètres de la page de Paramètres Directs et les paramètres fixes des Dispositifs. L'Annexe C récapitule les types d'erreur en transmission acyclique et l'Annexe D énumère les EventCodes (codes d'Evénements) (informations de diagnostic des Dispositifs). L'Annexe E spécifie les types disponibles de

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

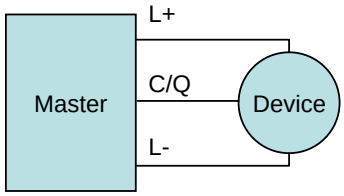
données de base et composées. L'Annexe F définit la structure des objets de Stockage des Données. L'Annexe G traite des exigences d'essai de conformité et de compatibilité électromagnétique et l'Annexe H donne des diagrammes de probabilité d'erreur résiduelle permettant de démontrer le niveau d'intégrité des données SDCI. L'Annexe I informative fournit un exemple de séquence de transmissions de données acycliques. L'Annexe J informative décrit deux méthodes recommandées de détection des modifications de paramètres dans le contexte du Stockage des Données.

5 Couche Physique (PL)

5.1 Généralités

5.1.1 Principes de base

Le système de connexion à 3 conducteurs de l'interface SDCI est fondé sur les spécifications de la CEI 60947-5-2. Les trois fils sont utilisés comme suit: (L+) pour l'alimentation 24 V, (L-) pour la masse, et (C/Q) pour le signal de commutation (Q) ou la communication SDCI (C), comme illustré en Figure 9.



Légende

Anglais	Français
Master	Maître
Device	Dispositif

Figure 9 – Système de connexion à trois conducteurs

NOTE 1 Les capteurs binaires conformes à la CEI 60947-5-2 sont compatibles avec le système de connexion à 3 conducteurs SDCI (y compris du point de vue de la consommation d'énergie).

La prise en charge du système de connexion à 3 conducteurs SDCI est obligatoire pour le Maître. Les ports présentant cette caractéristique sont appelés ports de classe A.

La classe de port A utilise un connecteur à quatre broches. Le quatrième conducteur peut être utilisé comme ligne de signal supplémentaire conformément à la CEI 61131-2. Sa prise en charge est facultative tant pour les Maîtres que pour les Dispositifs.

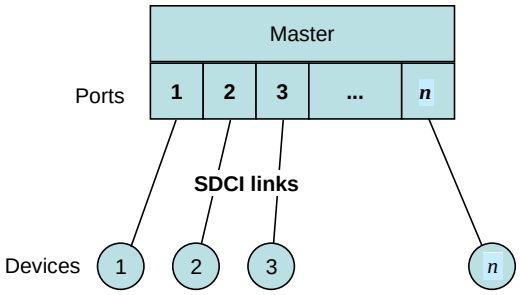
Des connexions à cinq conducteurs (classe de port B) sont spécifiées pour des Dispositifs nécessitant une puissance supplémentaire fournie à partir d'une alimentation 24 V indépendante (voir 5.5.1).

NOTE 2 Un dispositif de classe de port A utilisant le quatrième conducteur est incompatible avec un Maître de classe de port B.

5.1.2 Topologie

La topologie du système SDCI utilise des liaisons point à point entre un Maître et ses Dispositifs, comme illustré en Figure 10. Le Maître peut avoir plusieurs ports de connexion des Dispositifs. Un seul Dispositif doit être connecté à chaque port.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

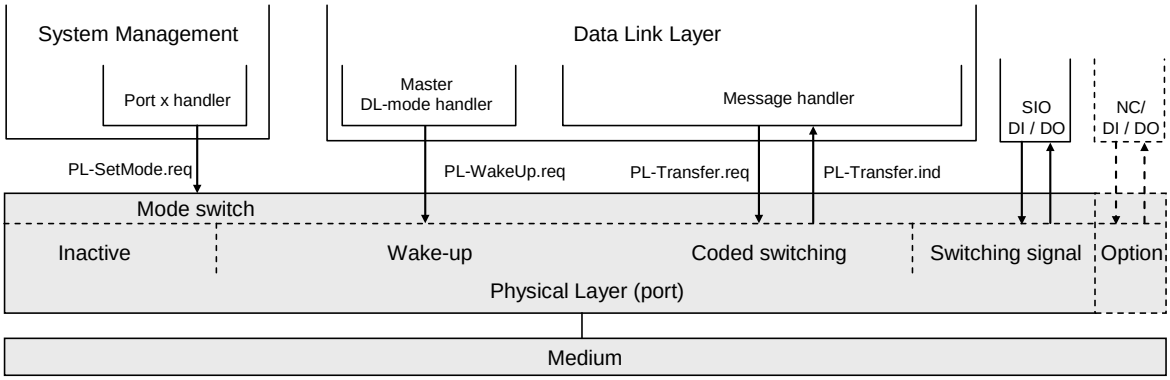
Anglais	Français
Master	Maître
SDCI links	Liaisons SDCI
Devices	Dispositifs

Figure 10 – Topologie de l'interface SDCI

5.2 Services de Couche Physique

5.2.1 Aperçu

La Figure 11 illustre une vue globale de la couche physique du Maître et de ses primitives de service.



Légende

Anglais	Français
System Management	Gestion Système
Data Link Layer	Couche de liaison de données
Port x handler	Module de traitement de Port-x
Master DL-mode handler	Module de traitement de mode DL du Maître
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Inactive	Inactif
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Option	Option
Physical Layer (port)	Couche Physique (port)
Medium	Support

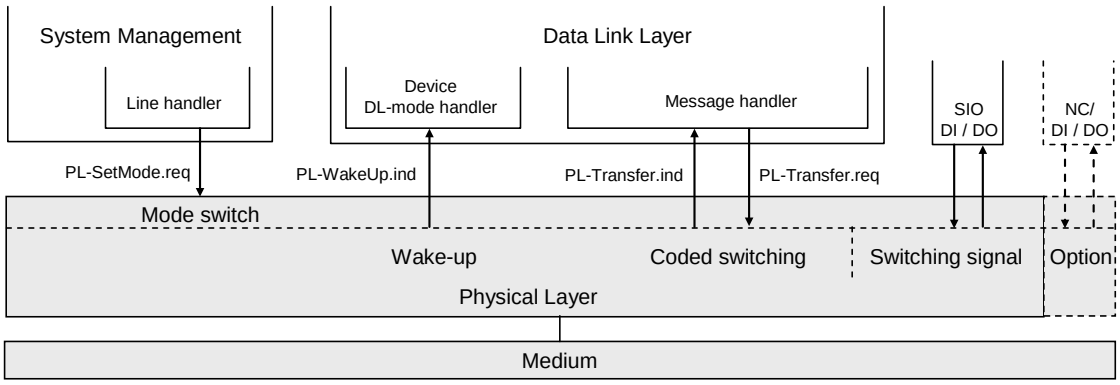
Figure 11 – Couche physique (Maître)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La couche physique définit le fonctionnement de la ligne C/Q en Figure 3, ainsi que le circuit d'attaque de la ligne (émetteur) et le récepteur correspondants d'un port particulier. Le Maître exploite cette ligne selon trois principaux modes (voir Figure 11): Inactif, "Signal de commutation" (DI/DO) ou "Commutation codée" (COMx). Le service PL-SetMode.req est chargé du passage dans l'un de ces modes.

Si le port est en mode inactif, la ligne C/Q doit avoir une haute impédance (flottante). En mode SIO, le port peut être utilisé comme une interface d'entrée ou de sortie normalisée conformément aux définitions de la CEI 61131-2 ou comme présenté dans le Tableau 6 respectivement. Les couches de communication de l'interface SDCI sont contournées comme illustré en Figure 11; les signaux sont directement traités au sein de l'application du Maître. En mode SDCI, le service PL_WakeUp.req génère une configuration de signal spéciale (impulsion de courant) qui peut être détectée par un Dispositif conforme SDCI connecté à ce port (voir 5.3.3.3).

La Figure 12 illustre une vue globale de la couche physique du Dispositif et de ses primitives de service.



Légende

Anglais	Français
System Management	Gestion Système
Data Link Layer	Couche de liaison de données
Line handler	Module de traitement de ligne
Device DL-mode handler	Module de traitement de mode DL du Dispositif
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Option	Option
Physical Layer	Couche physique
Medium	Support

Figure 12 – Couche physique (Dispositif)

La couche physique d'un Dispositif conforme à la Figure 12 respecte le même principe, sauf qu'il n'y a pas d'état inactif. Par défaut, à la mise sous tension ou lors du rebranchement du câble, le Dispositif doit fonctionner en mode SIO, comme une entrée numérique. Le Dispositif doit toujours pouvoir détecter une impulsion de courant de réactivation (demande de réactivation). Le service PL_WakeUp.ind rend compte d'une détection réussie de la demande de réactivation (généralement une interruption du microcontrôleur), qui est la condition exigée pour que le Dispositif passe en mode SDCI.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Une Commande du Maître MasterCommand spéciale (repli) envoyée par l'intermédiaire de l'interface SDCI ramène le Dispositif en mode SIO.

Par la suite, les services spécifiés sont ceux fournis par la PL à la Gestion Système et à la Couche de Liaison de Données (voir la Figure 83 et la Figure 94 pour un aperçu complet de tous les services). Le Tableau 1 énumère les attributions du Maître et du Dispositif quant à leur rôle d'initiateur ou de destinataire des services individuels de couche physique (PL).

Tableau 1 – Attributions des services du Maître et du Dispositif

Désignation du service	Maître	Dispositif
PL-SetMode	R	R
PL-WakeUp	R	I
PL-Transfer	I / R	R / I
Légende (voir 3.3.4) I Initiateur du service R Destinataire (Répondeur) du service		

5.2.2 Services de PL

5.2.2.1 PL_SetMode

Le service PL-SetMode est utilisé pour établir les caractéristiques électriques et les configurations de la Couche Physique. Les paramètres des primitives de service sont énumérés dans le Tableau 2.

Tableau 2 – PL_SetMode

Nom du paramètre	.req
Argument	M
TargetMode	M

Argument
Pour la demande de service, les paramètres spécifiques au service sont transmis dans l'argument.

TargetMode
Ce paramètre indique le mode de fonctionnement demandé.

Valeurs autorisées:
INACTIVE (impédance haute de la ligne C/Q),
DI (ligne C/Q en mode d'entrée numérique),
DO (ligne C/Q en mode de sortie numérique),
COM1 (ligne C/Q en mode COM1),
COM2 (ligne C/Q en mode COM2),
COM3 (ligne C/Q en mode COM3)

5.2.2.2 PL_WakeUp

Le service PL-WakeUp lance ou indique une séquence spécifique qui prépare la Couche Physique à envoyer et à recevoir des demandes de communication (voir 5.3.3.3). Ce service non confirmé n'a pas de paramètres. Son aboutissement avec succès ne peut être vérifié que par un Maître lors d'une tentative de communication avec le Dispositif. Les primitives de service sont énumérées dans le Tableau 3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 3 – PL_WakeUp

Nom du paramètre	.req	.ind
<Aucun>		

5.2.2.3 PL_Transfer

Le service PL-Transfer est utilisé pour l'échange de données SDCI entre la Couche de Liaison de Données et la Couche Physique. Les paramètres des primitives de service sont énumérés dans le Tableau 4.

Tableau 4 – PL_Transfer

Nom du paramètre	.req	ind.
Argument		
Données	M	M
Résultat(+)		S
Résultat (-)		S
État		M

Argument
Pour la demande de service, les paramètres spécifiques au service sont transmis dans l'argument.

Données
Ce paramètre contient la valeur des données transférées sur l'interface SDCI.

Valeurs autorisées: 0…255

Résultat (+):
Ce paramètre de sélection indique que la demande de service a été exécutée avec succès.

Résultat (-):
Ce paramètre de sélection indique que la demande de service a échoué.

État
Ce paramètre contient des informations supplémentaires concernant l'état du transfert.

Valeurs autorisées:
PARITY_ERROR (l'UART a détecté une erreur de parité),
FRAMING_ERROR (détection de bit d'arrêt UART non valide),
OVERRUN (collision d'octets dans l'UART)

5.3 Émetteur/Récepteur

5.3.1 Méthode de description

La couche physique est spécifiée au moyen d'exigences électriques et temporelles. Les exigences électriques définissent séparément des niveaux de signaux et de courants pour le Maître et le Dispositif sous la forme de schémas de référence. Les exigences temporelles définissent le processus de transmission des signaux (spécifiquement pour le récepteur) et une fonction spéciale de détection des signaux.

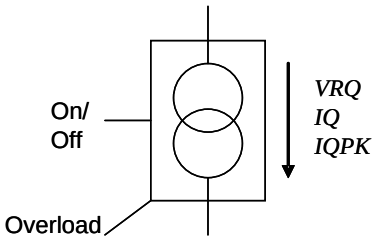
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

5.3.2 Exigences électriques

5.3.2.1 Généralités

Le circuit d'attaque de la ligne est défini par un schéma de référence correspondant à la Figure 13. Du côté du Maître, un émetteur comprend une combinaison de deux circuits d'attaque et d'un collecteur de courant. Du côté du Dispositif, sous sa forme la plus simple, l'émetteur se présente sous la forme d'un circuit d'attaque à commutation p. Il peut y avoir en option un circuit d'attaque supplémentaire à commutation n ou sans commutation (ceci permet également d'opter pour un fonctionnement en configuration de sortie symétrique).

À l'état opérationnel ON (Actif), les variables descriptives sont la tension résiduelle *VRQ*, le courant d'attaque normalisé *IQ* et le courant de crête *IQPK*. La source est commandée par le signal On/Off (Actif/Non actif). Un événement de courant de surcharge est indiqué à la sortie "Surcharge" (OVD). Cette fonctionnalité peut être utilisée pour la détection de l'impulsion de courant (de réactivation).



Légende

Anglais	Français
On/Off	Marche/Arrêt
Overload	Surcharge

Figure 13 – Schéma de référence du circuit d'attaque de la ligne

Le récepteur est défini par un schéma de référence comme illustré en Figure 14. Il assure la fonction d'un comparateur et il est spécifié par ses seuils de commutation *VTH* et une hystérésis *VHYS* entre les seuils de commutation. La sortie indique le niveau logique (Haut ou Bas (H/L)) à l'entrée du récepteur.

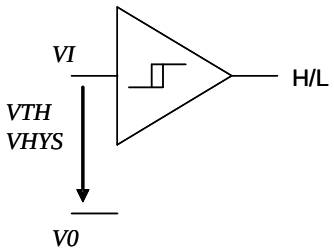
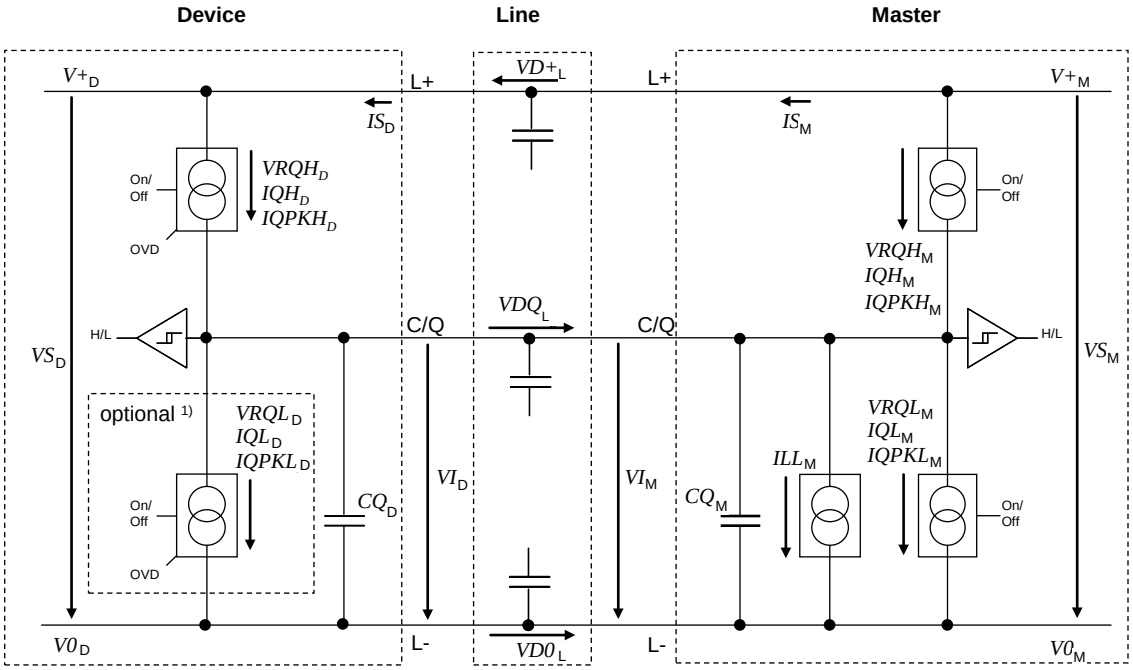


Figure 14 – Schéma de référence du récepteur

La Figure 15 illustre le schéma de référence de l'interconnexion du Maître et du Dispositif pour le système de connexion à 3 conducteurs SDCI.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



1) facultatif: circuit d'attaque côté bas (en configuration symétrique uniquement)

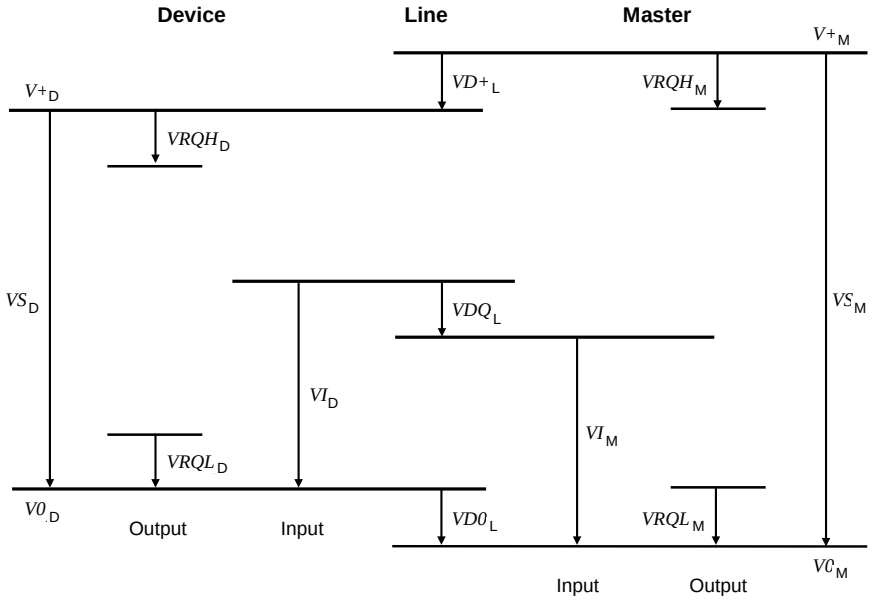
Légende

Anglais	Français
Device	Dispositif
Line	Ligne
Master	Maître
On/Off	Marche/Arrêt
H/L	H/L (haut/bas)

Figure 15 – Schéma de référence pour le système de connexion à 3 conducteurs SDCI

Les illustrations et tableaux de paramètres suivants font référence aux définitions des niveaux de tension donnés en Figure 16. Les index des paramètres font référence au Maître (M), au Dispositif (D) ou à la ligne (L). Les chutes de tension sur $VD+_L$, VDQ_L et $VD0_L$ de la ligne sont implicitement spécifiées en 5.5 par le biais des paramètres du câblage.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device	Dispositif
Line	Ligne
Master	Maître
Output	Sortie
Input	Entrée

Figure 16 – Définition des niveaux de tension

5.3.2.2 Récepteur

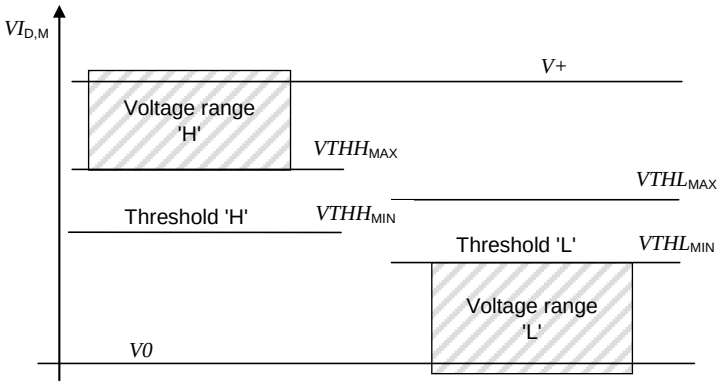
La définition des plages de tension et des seuils de commutation est la même pour le Maître et le Dispositif. Les définitions du Tableau 5 s'appliquent.

Tableau 5 – Caractéristiques électriques d'un récepteur

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
$VTHH_{D,M}$	Seuil d'entrée 'H'	10,5	n/a	13	V	Voir la NOTE 1
$VTHL_{D,M}$	Seuil d'entrée 'L'	8	n/a	11,5	V	Voir la NOTE 1
$VHYS_{D,M}$	Hystérésis entre seuils d'entrée 'H' et 'L'	0	n/a	n/a	V	Ne doit pas être négative Voir la NOTE 2
$VIL_{D,M}$	Plage de tension admissible 'L'	$V0_{D,M}$ -1,0	n/a	n/a	V	En référence à la tension d'alimentation négative pertinente
$VIH_{D,M}$	Plage de tension admissible 'H'	n/a	n/a	$V+_{D,M}$ + 1,0	V	En référence à la tension d'alimentation positive pertinente
NOTE 1 Les seuils sont compatibles avec la définition des entrées numériques de type 1 données dans la CEI 61131-2. NOTE 2 Tension d'hystérésis $VHYS = VTHH - VTHL$.						

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Figure 17 montre les seuils de commutation pour la détection des signaux Bas et Haut.



Légende

Anglais	Français
Voltage range	Plage de tension
Threshold	Seuil

Figure 17 – Seuils de commutation

5.3.2.3 Port du Maître

Les définitions du Tableau 6 sont valides pour les caractéristiques électriques d'un port du Maître.

Tableau 6 – Caractéristiques électriques d'un port du Maître

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
VS_M	Tension d'alimentation des Dispositifs	20	24	30	V	Voir la Figure 16
IS_M	Courant d'alimentation des Dispositifs	200	n/a	n/a	mA	Alimentation externe requise pour > 200 mA
$ISIR_M$	Capacité d'impulsion de courant pour des Dispositifs	400	n/a	n/a	mA	La capacité du courant d'alimentation du Maître pour une durée minimale de 50 ms à 18 V après mise sous tension de l'alimentation du port
ILL_M	Courant de charge ou de décharge pour 0 V < VI_M < 5 V 5 V < VI_M < 15 V 15 V< VI_M < 30 V	 0 5 5	 n/a n/a n/a	 15 15 15	 mA mA mA	Voir la NOTE 1
$VRQH_M$	Tension résiduelle 'H'	n/a	n/a	3	V	Chute de tension relativement à $V+_M$, au courant d'attaque maximal IQH_M
$VRQL_M$	Tension résiduelle 'L'	n/a	n/a	3	V	Chute de tension relativement à $V0_M$, au courant d'attaque maximal IQL_M
IQH_M	Courant d'attaque CC 'H'	100	n/a	n/a	mA	

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
$IQPKH_M$	Courant crête de sortie 'H'	500	n/a	n/a	mA	Valeur absolue Voir la NOTE 2
IQL_M	Courant d'attaque CC 'L'	100	n/a	n/a	mA	
$IQPKL_M$	Courant crête de sortie 'L'	500	n/a	n/a	mA	Valeur absolue Voir la NOTE 2
CQ_M	Capacité d'entrée	n/a	n/a	1,0	nF	f= 0 MHz à 4 MHz
NOTE 1 Les courants sont compatibles avec la définition des entrées numériques de type 1 données dans la CEI 61131-2. Toutefois, pour la plage 5 V < VI_M < 15 V, le courant minimum est de 5 mA iau lieu de 2 mA afin d'obtenir des taux de dérive suffisamment faibles pour des Dispositifs strictement de commutation p. NOTE 2 Courant de demande de réactivation (5.3.3.3).						

5.3.2.4 Dispositif

Les définitions du Tableau 7 sont valides pour les caractéristiques électriques d'un Dispositif.

Tableau 7 – Caractéristiques électriques d'un Dispositif

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
VS_D	Tension d'alimentation	18	24	30	V	Voir la Figure 16
ΔVS_D	Ondulation	n/a	n/a	1,3	V_{pp}	Les limites de valeur absolue crête à crête ne doivent pas être dépassées. $f_{ondulation}$= CC à 100 kHz
$VRQH_D$	Tension résiduelle 'H'	n/a	n/a	3	V	Chute de tension relativement à $V+_D$ (CEI 60947-5-2)
$VRQL_D$	Tension résiduelle 'L'	n/a	n/a	3	V	Chute de tension relativement à $V0_D$
IQH_D	Courant d'attaque CC Sortie de commutation P (état "Actif")	50	n/a	minimal($IQPKL_M$)	mA	Valeur minimale due au repli vers l'entrée numérique conformément à la CEI 61131-2, type 2
IQL_D	Courant d'attaque CC Sortie de commutation N (état "Actif")	0	n/a	minimal ($IQPKH_M$)	mA	Uniquement pour des étages de sortie en configuration symétrique
IQQ_D	Courant passif à VO_D (état "Non actif")	0	n/a	15	mA	Courant de polarisation ou résiduel avec étages du circuit d'attaque des sorties désactivés
CQ_D	Capacité d'entrée	0	n/a	1,0	nF	Capacité effective entre C/Q et L+ ou L- du Dispositif à l'état de réception

La valeur de 1 nF est applicable pour une vitesse de transmission de 230,4 kbit/s. La capacité d'entrée CQ_D peut être élargie à un niveau maximum de 10 nF dans le cas d'une conception d'étage en configuration symétrique lors d'un fonctionnement à des vitesses de transmission

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

inférieures, sous réserve de satisfaire à toutes les exigences concernant les paramètres dynamiques spécifiées au 5.3.3.2.

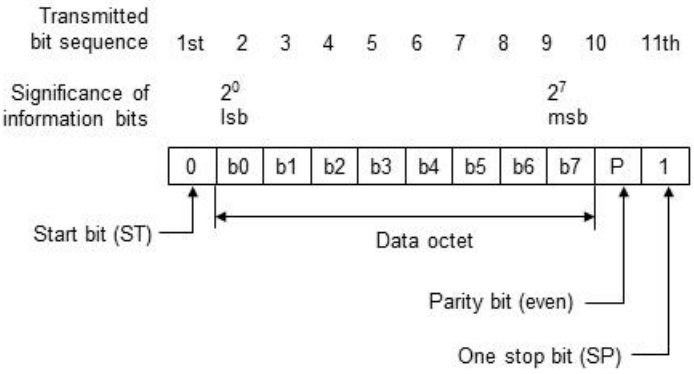
5.3.3 Exigences temporelles

5.3.3.1 Méthode de transmission

La modulation “NRZ” (Non Return to Zero – non retour à zéro) est utilisée pour le codage bit par bit. Une valeur logique “1” correspond à une différence de potentiel de 0 V entre C/Q et L-. Une valeur logique “0” correspond à une différence de potentiel de +24 V entre C/Q et L-.

Le niveau de circuit ouvert sur la ligne C/Q est 0 V en référence à L-. Un bit de départ a une valeur logique “0”, c'est-à-dire +24 V en référence à L-.

Une trame UART est utilisée pour le codage "octet de données" par "octet de données". Le format de la trame UART de l'interface SDCI est une chaîne binaire structurée comme illustré en Figure 18.



Légende:
lsb bit de poids faible
msb bit de poids fort

Légende

Anglais	Français
Transmitted bit sequence	Séquence binaire transmise
1st 2 3 4 5 6 7 8 9 10 11th	1er 2 3 4 5 6 7 8 9 10 11ème
Significance of information bits	Poids des bits
Start bit (ST)	Bit de départ (ST)
Data octet	Octet de données
Parity bit (even)	Bit de parité (paire)
One stop bit (SP)	Un bit d'arrêt (SP)

Figure 18 – Format d'une trame UART de l'interface SDCI

La définition du format de la trame UART repose sur l'ISO 1177 et l'ISO/CEI 2022.

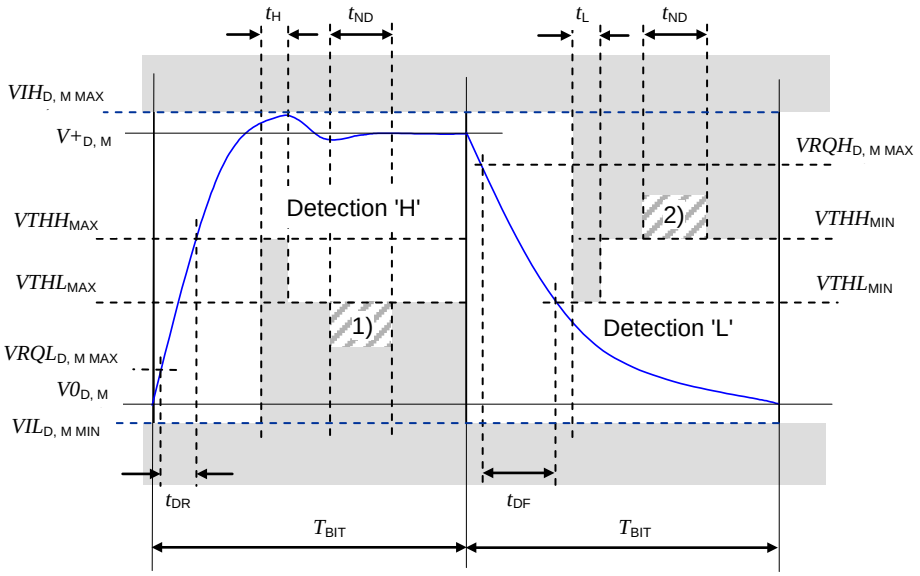
5.3.3.2 Caractéristiques de transmission

Les caractéristiques temporelles de transmission sont présentées sous la forme d'un diagramme en œil avec les plages de signaux admissibles (voir la Figure 19). Ces plages sont applicables au récepteur à la fois dans le Maître et dans le Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Quelles que soient les conditions aux limites, l'émetteur doit générer sur la connexion C/Q du récepteur une caractéristique de tension qui s'inscrit dans la plage admissible du diagramme en œil.

Le récepteur doit détecter des bits comme une forme de signal valide dans la plage admissible du diagramme en œil sur la connexion C/Q. Les formes de signal dans les zones de “non-détection” (en dessous de $VTHL_{MAX}$ ou au-dessus de $VTHH_{MIN}$ et au sein de t_{ND}) ne doivent pas donner lieu à des bits non valides.



Légende

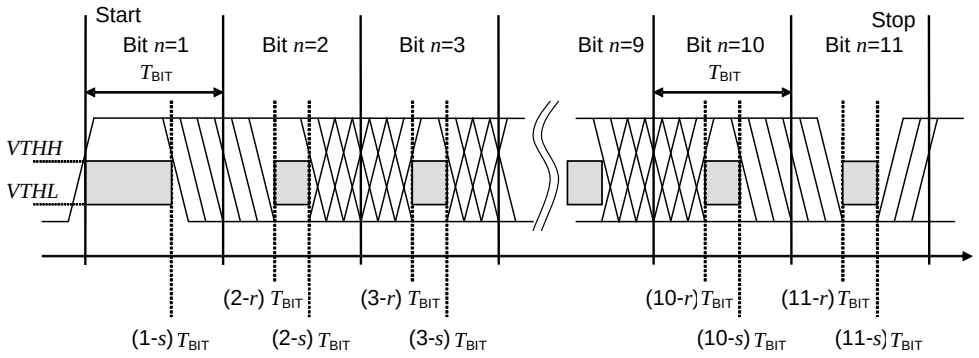
Anglais	Français
Detection	Détection

NOTE Dans la figure, 1) = pas de détection "L"; et 2) = pas de détection "H"

Figure 19 – Diagramme en oeil pour la détection de 'H' et 'L'

Pour qu'une trame UART soit correctement détectée, une caractéristique de signal, telle que présentée en Figure 20, est nécessaire du côté récepteur. Le délai de propagation du signal entre le signal C/Q et l'entrée UART doit être pris en compte. Le temps T_{BIT} indique toujours le débit binaire du récepteur.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Start bit	Bit de départ
Stop bit	Bit d'arrêt

Figure 20 – Diagramme en oeil pour la détection correcte d'une trame UART

Pour chaque bit n dans la séquence binaire (n =1...11) d'une trame UAR, le temps $(n\text{-}r)T_{BIT}$ (voir le Tableau 8 pour les valeurs de r) désigne le délai au bout duquel un niveau correct doit être atteint dans les plages 'H' ou 'L' comme présenté dans le diagramme en œil de la Figure 19. Le temps $(n\text{-}s)$ $TBIT$ (voir le Tableau 8 pour les valeurs de s) décrit la durée qui doit s'écouler entre les changements de niveaux. On doit toujours faire référence au diagramme en œil de la Figure 19, lorsque les caractéristiques de signaux dans un temps binaire donné sont concernées.

Cette représentation permet une pondération variable des paramètres d'influence du récepteur, tels que:"précision de la vitesse de transmission", "distorsion de largeur binaire" et "taux de dérive" du récepteur.

Le Tableau 8 présente les caractéristiques dynamiques de la transmission.

Tableau 8 – Caractéristiques dynamiques de la transmission

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
f_{DTR}	Vitesse de transmission	n/a	4,8 38,4 230,4	n/a	kbit/s	COM1 COM2 COM3
T_{BIT}	Temps binaire à 4,8 kbit/s à 38,4 kbit/s à 230,4 kbit/s		 208,33 26,04 4,34		 µs µs µs	
Δf_{DTRM}	Précision de la vitesse de transmission du Maître à 4,8 kbit/s à 38,4 kbit/s à 230,4 kbit/s	 -0,1 -0,1 -0,1	 n/a n/a n/a	 +0,1 +0,1 +0,1	 % % %	Tolérance sur la vitesse de transmission du Maître $\Delta T_{BIT}/T_{BIT}$
r	Début de la période de détection dans un bit donné en référence au front montant du bit de départ	0,65	n/a	n/a	-	Calculé dans chaque cas à partir de la fin d'un bit à un taux d'échantillonnage UART de 8

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
s	Fin de la période de détection dans un bit donné en référence au front montant du bit de départ	n/a	n/a	0,22	-	Calculé dans chaque cas à partir de la fin d'un bit à un taux d'échantillonnage UART de 8
T_{DR}	Temps de croissance à 4,8 kbit/s à 38,4 kbit/s à 230,4 kbit/s	0 0 0 0	n/a n/a n/a n/a	0,20 41,7 5,2 869	T_{BIT} µs µs ns	En référence à l'unité de temps binaire
t_{DF}	Temps de décroissance à 4,8 kbit/s à 38,4 kbit/s à 230,4 kbit/s	0 0 0 0	n/a n/a n/a n/a	0,20 41,7 5,2 869	T_{BIT} µs µs ns	En référence à l'unité de temps binaire
t_{ND}	Temps de suppression du bruit	n/a	n/a	1/16	T_{BIT}	Durée admissible d'un signal de réception au-dessus / en dessous du seuil de détection sans qu'il n'y ait de détection
t_H	Temps de détection niveau Haut	1/16	n/a	n/a	T_{BIT}	Durée d'un signal de réception au-dessus du seuil de détection pour un niveau 'H'
t_L	Temps de détection niveau Bas	1/16	n/a	n/a	T_{BIT}	Durée d'un signal de réception au-dessous du seuil de détection pour un niveau 'H'

Les paramètres 'r' et s' s'appliquent du côté récepteur Maître ou Dispositif, respectivement. Cette définition permet de spécifier de manière plus souple la précision de l'oscillateur, la distorsion binaire et le taux de dérive du côté Dispositif. La distorsion de largeur-binaire globale sur le dernier bit de la trame UART doit fournir un niveau correct dans la plage illustrée en Figure 20.

5.3.3.3 Impulsion de courant de réactivation

La fonction de réactivation est utilisée pour demander à un Dispositif de passer en mode COMx.

Une invocation de service (PL_WakeUp.req) à partir de la couche DL lance le processus de réactivation (voir 5.2.2.2).

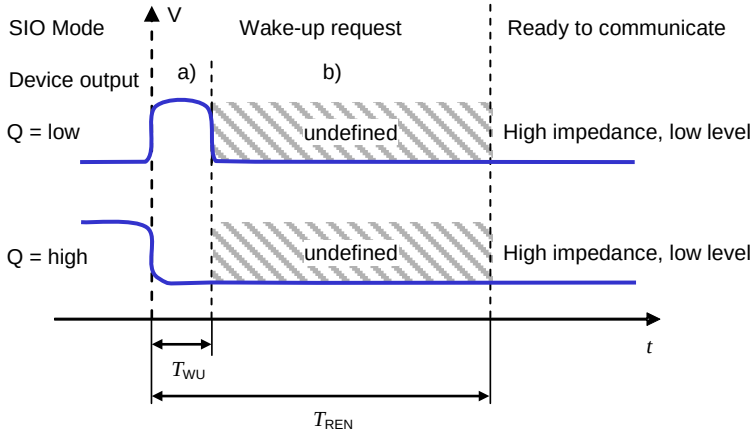
La demande de réactivation (WURQ) commence par une impulsion de courant induite par le Maître (port) pendant une durée T_{WU}. La demande de réactivation comporte les phases suivantes (voir la Figure 21):

a) Injection d'un courant IQ_{WU} par le Maître en fonction du niveau de la connexion C/Q. Pour un signal d'entrée équivalent au niveau logique "1", il s'agit d'une source de courant; pour un signal d'entrée équivalent au niveau logique "0", il s'agit d'un collecteur de courant.

b) Temporisation du Dispositif jusqu'à ce qu'il soit prêt à recevoir.

L'impulsion de demande de réactivation peut être détectée par le Dispositif du fait d'une modification de la tension sur la ligne C/Q ou par évaluation du courant de l'élément du circuit d'attaque correspondant dans le temps T_{WU}. Des exemples de Dispositifs à faible puissance de sortie sont illustrés en Figure 21.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
SIO Mode	Mode SIO
Wake-up request	Demande de réactivation
Ready to communicate	Prêt à communiquer
Device output	Sortie de dispositif
Q = low	Q = bas
undefined	indéfini
High impedance, low level	Haute impédance, niveau bas
Q = high	Q = haut

Figure 21 – Demande de réactivation

Le Tableau 9 définit les propriétés du courant et des temporisations associées à la demande de réactivation. Pour les valeurs de $IQPKL_M$ et $IQPKH_M$, voir le Tableau 6.

Tableau 9 – Caractéristiques de la demande de réactivation

Proprété	Designation	Minimale	Type	Maximale	Unité	Remarque
IQ_{WU}	Amplitude de l'impulsion de courant de réactivation du Maître	$IQPKL_M$ ou $IQPKH_M$	n/a	n/a	mA	Impulsion de courant suivie par l'état de commutation du Dispositif
T_{WU}	Durée de l'impulsion de courant de réactivation du Maître	75	n/a	85	µs	Propriété du Maître
T_{REN}	Délai d'activation de la réception	n/a	n/a	500	µs	Propriété du Dispositif

5.4 Alimentation

5.4.1 Options d'alimentation

Le système de connexion SDCI dispose de lignes d'alimentation dédiées, outre la ligne de signalisation. La partie communication d'un Dispositif doit toujours être alimentée par le Maître en utilisant les lignes d'alimentation définies dans le système de connexion à 3 conducteurs (Power1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le courant d'alimentation maximal disponible à partir d'un port du Maître est spécifié dans le Tableau 6.

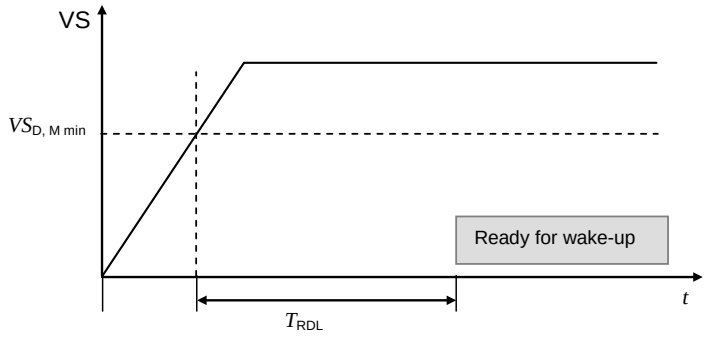
La partie application du dispositif peut être alimentée de trois manières différentes:

- par l'intermédiaire des lignes d'alimentation du système de connexion à 3 conducteurs SDCI (ports de classe A), en utilisant Power1;
- à partir de lignes d'alimentation supplémentaires du système de connexion à 5 conducteurs SDCI (ports de classe B), en utilisant une source d'alimentation supplémentaire au niveau du Maître (Power2);
- à partir d'une alimentation locale au niveau du Dispositif (spécifique à la conception).

La classe de port A permet une consommation d'énergie allant jusqu'à 200 mA, comme spécifié dans le Tableau 6. La consommation d'énergie maximale sur les ports de classe B dépend de la méthode de connexion sélectionnée. M12 permet uniquement d'aller jusqu'à 3,5 A supplémentaires.

5.4.2 Exigences de mise sous tension

La Figure 22 illustre comment le comportement à la mise sous tension d'un Dispositif est défini par le temps de montée en puissance de l'alimentation Power1 et par le temps interne du Dispositif afin qu'il soit prêt pour l'opération de réactivation.



Légende

Anglais	Français
Ready for wake-up	Prêt pour réactivation

Figure 22 – Temporisation de la mise sous tension pour Power1

À la mise sous tension, il est obligatoire qu'un Dispositif atteigne l'état prêt à la réactivation dans les limites temporelles spécifiées dans le Tableau 10.

Tableau 10 – Temporisation de la mise sous tension

Propriété	Designation	Minimale	Type	Maximale	Unité	Remarque
T_{RDL}	Prêt à la réactivation suite à la mise sous tension	n/a	n/a	300	ms	Temps de montée en puissance du Dispositif jusqu'à ce qu'il soit prêt à détecter le signal de réactivation (voir la NOTE)
NOTE Equivalent au délai avant disponibilité de la CEI 60947-5-2.						

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

5.5 Support

5.5.1 Connecteurs

La connectique et l'affectation des broches du Maître et du Dispositif se fondent sur les spécifications de la CEI 60947-5-2, avec les extensions définies dans les alinéas ci-dessous. La classe de port A utilise au maximum quatre broches des connecteurs M5, M8 et M12. La classe de port B utilise uniquement les connecteurs M12 à 5 broches. Ces connecteurs M12 sont mécaniquement codés A conformément à la CEI 61076-2-101.

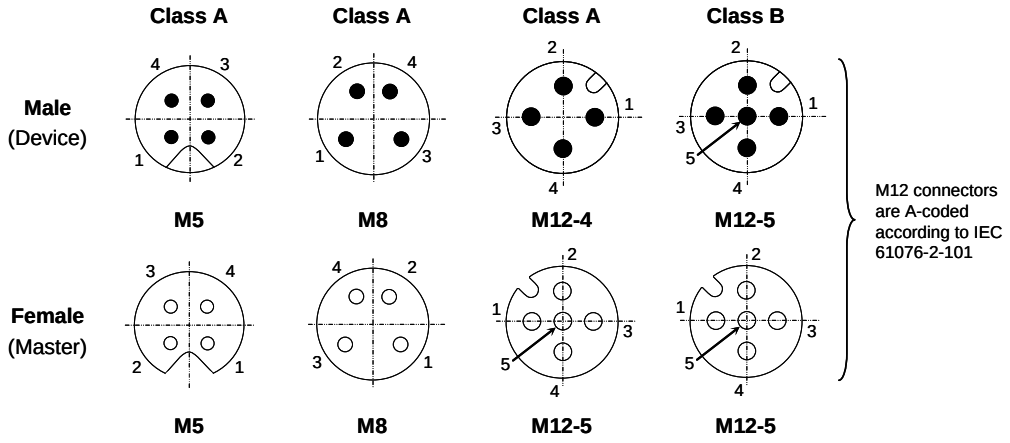
NOTE Pour des raisons historiques ou de compatibilité, il est également possible d'utiliser un câblage direct ou différents types de connecteurs, à condition qu'ils ne portent pas préjudice aux caractéristiques électriques et qu'ils emploient la désignation de signaux spécifiée dans la présente norme.

Les connecteurs femelles sont attribués au Maître et les connecteurs mâles au Dispositif. Le Tableau 11 donne l'affectation des broches et la Figure 23 illustre l'agencement et le codage mécanique des connexions M12, M8 et M5.

Tableau 11 – Affectation des broches

Broche	Signal	Désignation	Remarque
1	L+	Alimentation (+)	Voir le Tableau 7
2	I/Q P24	NC/DI/DO (classe de port A) P24 (classe de port B)	Option 1: NC (non connecté) Option 2: DI Option 3: DI, puis DO configurées Option 4: Alimentation supplémentaire des Dispositifs alimentés (classe de port B)
3	L-	Alimentation (-)	Voir le Tableau 7
4	C/Q	SIO/SDCI	Mode E/S normalisé (DI/DO) ou SDCI (voir les caractéristiques électriques de DO dans le Tableau 6).
5	NC N24	NC (classe de port A) N24 (classe de port B)	Option 1: Ne doit pas être connecté du côté Maître (classe de port A). Option 2: Référence à l'alimentation supplémentaire (classe de port B)
NOTE Le connecteur M12 a toujours 5 broches du côté Maître (femelle).			

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

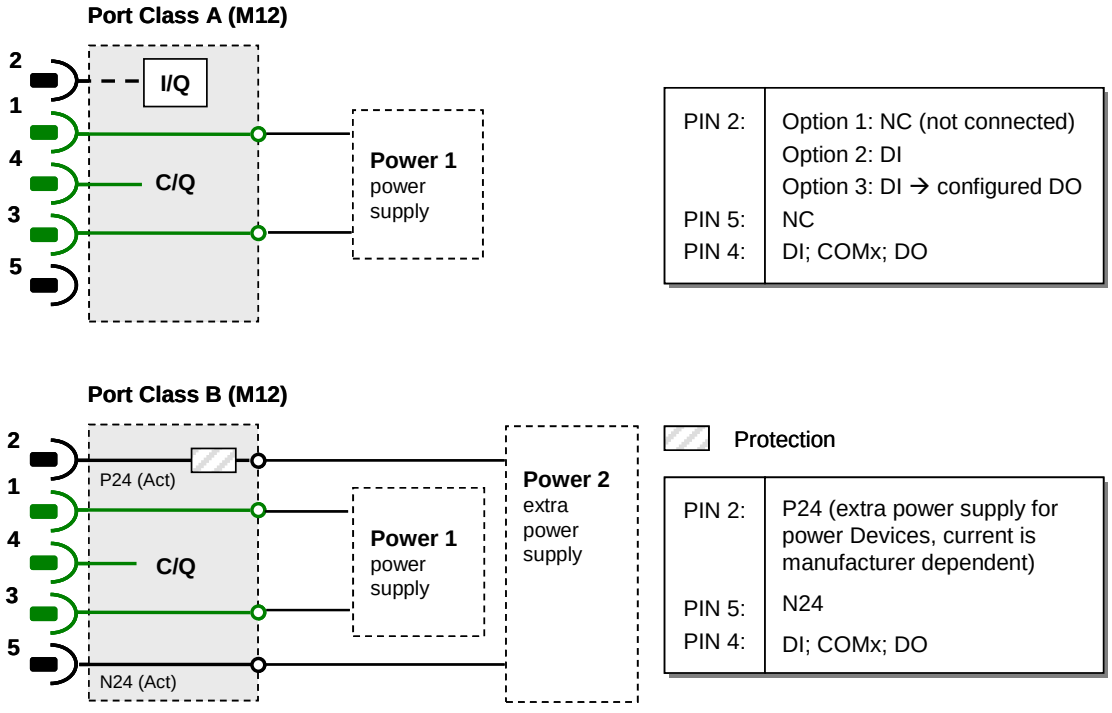


Légende

Anglais	Français
Class	Classe
Male (Device)	Mâle (Dispositif)
Female (Master)	Femelle (Maître)
M12 connectors are A- coded according to IEC 61076-2-101	Ces connecteurs M12 sont codés A conformément à la CEI 61076-2-101

Figure 23 – Vue de face de l'agencement des broches

La Figure 24 illustre l'agencement des deux classes de port, A et B. Afin d'éviter les éventuels risques d'incompatibilité, les ports de classe B doivent être marqués de manière à faire la distinction avec les ports de classe A.



Légende

Anglais	Français
Port Class	Classe de port

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Power supply	Alimentation
PIN	Broche
Option 1: NC (not connected)	Option 1: NC (non connecté)
Option 3: DI configured DO	Option 3: DI--> DO configuré
extra power supply	Alimentation supplémentaire
P24 (extra power supply for power Devices, current is manufacturer dependent)	P24 (alimentation supplémentaire pour dispositifs alimentés, le courant dépend du fabricant)

Figure 24 – Définitions des ports de classes A et B

5.5.2 Câblage

Pour la communication SDCI, le support de transmission est un câble à plusieurs conducteurs (3 ou plus). Les définitions des alinéas suivants couvrent implicitement les définitions de la tension statique du Tableau 5 et de la Figure 16. Pour assurer la fiabilité fonctionnelle, les propriétés du câblage doivent être conformes au Tableau 12.

Tableau 12 – Caractéristiques du câblage

Propriété	Minimale	Type	Maximale	Unité
Longueur	0	n/a	20	m
Résistance de boucle globale RL_{eff}	n/a	n/a	6,0	Ω
Capacité effective de la ligne CL_{eff}	n/a	n/a	3,0	nF (<1 MHz)

La résistance de boucle RL_{eff} et la capacité effective de la ligne CL_{eff} peuvent être mesurées comme illustré en Figure 25.

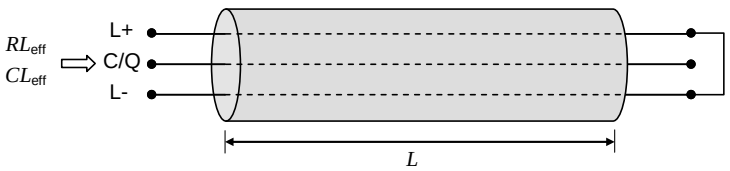


Figure 25 – Schéma de référence de la capacité effective de la ligne et de la résistance de boucle

Le Tableau 13 présente les conducteurs du câble et les codes de couleurs qui leur sont attribués.

Tableau 13 – Affectation des conducteurs du câble

Signal	Désignation	Couleur	Remarque
L-	Alimentation (-)	Bleu [a]	Système de connexion SDCI à 3 conducteurs
C/Q	Signal de communication	Noir [a]	Système de connexion SDCI à 3 conducteurs
L+	Alimentation (+)	Marron [a]	Système de connexion SDCI à 3 conducteurs
I/Q	DI ou DO	Blanc [a]	Facultatif
P24	Alimentation supplémentaire (+)	N'importe quelle autre	Facultatif
N24	Alimentation supplémentaire (-)	N'importe quelle autre	Facultatif
[a] Correspondant à la CEI 60947-5-2			

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

6 Entrées et sorties normalisées (SIO)

La Figure 83 et la Figure 94 montrent comment le mode SIO permet à un Dispositif de contourner les couches de communication SDCI et de mapper le signal DI ou DO directement dans le message d'échange des données du bus de terrain ou système de niveau supérieur. Le changement entre le mode SDCI et le mode SIO est défini par la configuration de l'utilisateur ou implicitement par les services des applications du Maître. La Gestion Système prend en charge l'initialisation correspondante ou la désactivation des couches de communication SDCI et de la couche physique (mode commutation). Les caractéristiques des interfaces pour les signaux DI et DO sont issues de la CEI 61131-2, type 1.

7 Couche de liaison de données (DL)

7.1 Généralités

Les couches de liaison de données de l'interface SDCI sont chargées de la transmission des messages entre un Maître et un Dispositif, par l'intermédiaire de la liaison physique. La couche de liaison de données utilise plusieurs types de séquences M ("séquence de message") pour différentes catégories de données.

Un ensemble de services DL est utilisé par la couche d'application (AL) pour l'échange de Données de Processus (PD) et de Données à la Demande (OD). Un autre ensemble de services DL est utilisé par la Gestion Système (SM) pour la récupération des paramètres d'identification de Dispositif et le réglage des diagrammes d'états dans la couche DL. La couche DL utilise des services PL pour commander la couche physique (PL) et échanger des trames UART. La couche DL est chargée de la détection d'erreurs dans les messages (qu'elles soient internes ou renvoyées par la PL) et de prendre les mesures correctives appropriées (par exemple une nouvelle tentative).

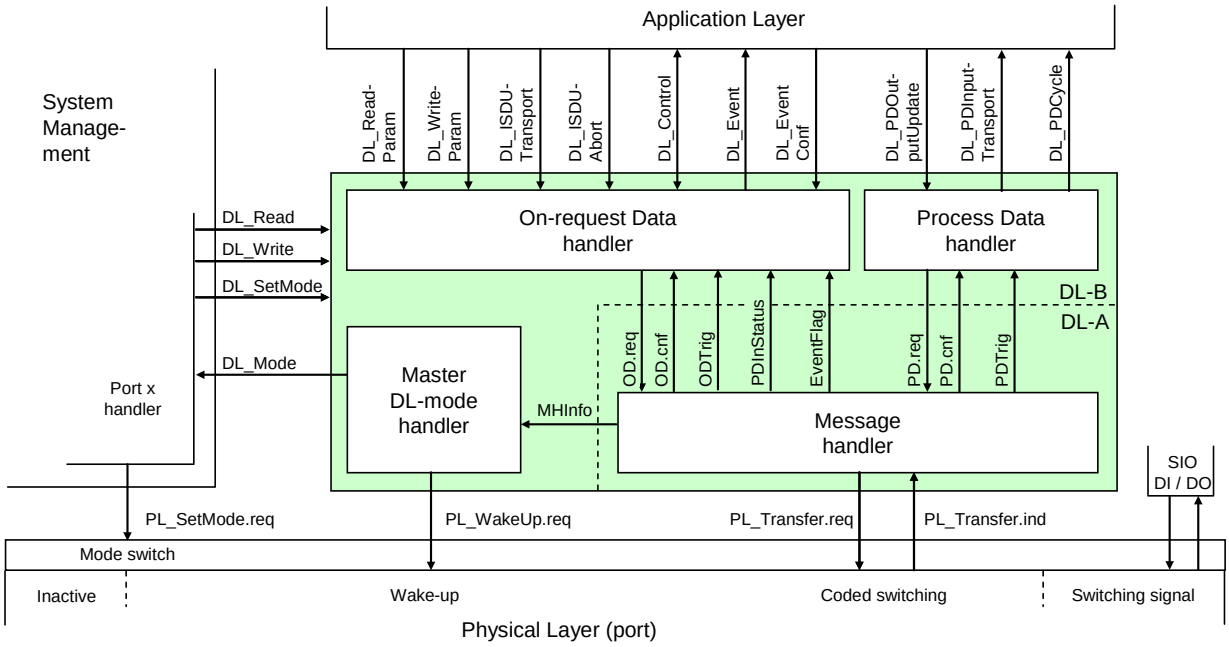
Les couches de liaison de données sont structurées en fonction de la nature des catégories de données en modules de traitement de Données de Processus et modules de traitement de Données à la Demande, qui à leur tour utilisent un module de traitement de messages pour traiter les transmissions de messages demandées. Les modes spéciaux des ports du Maître, tels que réactivation, COMx ainsi que SIO (désactivation de la communication), nécessitent un module de traitement de mode DL dédié au sein de la DL du Maître. La modulation du signal spécial de réactivation nécessite une détection de signaux du côté du Dispositif et de ce fait, elle utilise un module de traitement de mode DL au sein de la DL du Dispositif. Chaque module de traitement dispose de son propre diagramme d'états.

La couche de liaison de données est divisée en section A de DL avec ses propres services internes, et section B de DL avec des services externes.

La couche DL utilise des invocations supplémentaires de procédures réglementaires internes entre les modules de traitement définies dans la section "éléments internes" des tableaux de transitions d'états correspondants.

La Figure 26 illustre une vue globale de la structure et des services de la couche de liaison de données du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

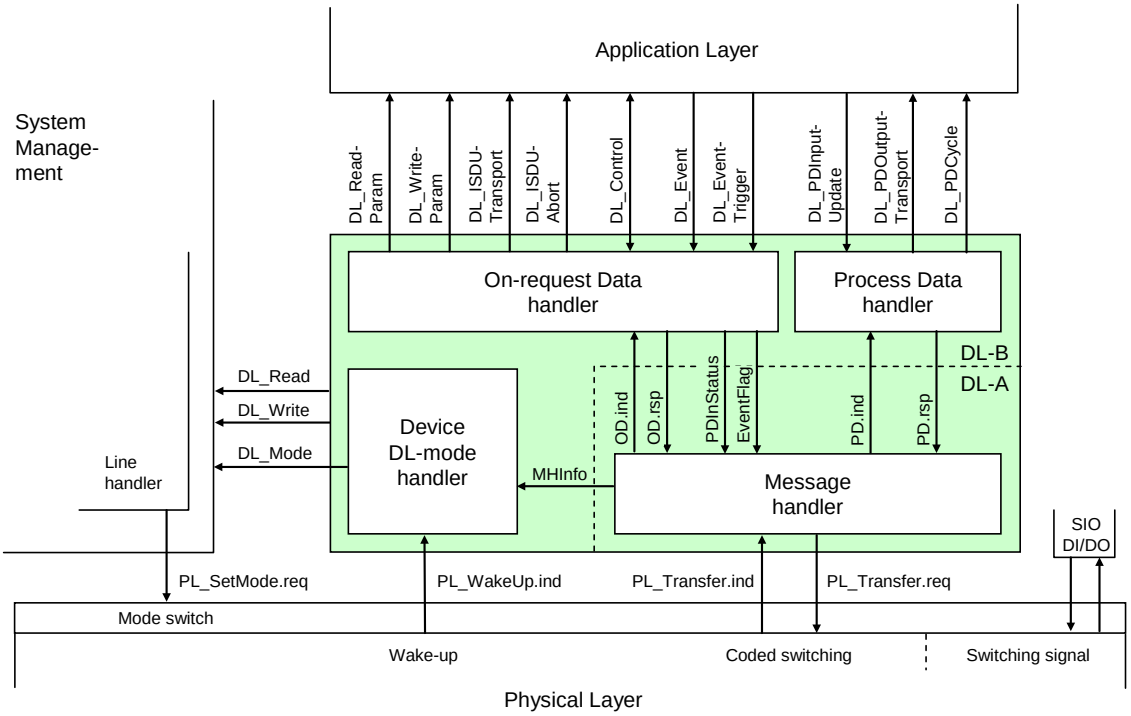
Anglais	Français
System Management	Gestion Système
Application Layer	Couche d'Application
On-request Data handler	Module de traitement de Données à la Demande
Process Data handler	Module de traitement des Données de Processus
Port x handler	Module de traitement de Port-x
Master DL-mode handler	Module de traitement de mode DL du Maître
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Inactive	Inactive
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer (port)	Couche Physique (port)

NOTE Cette figure utilise les conventions données en 3.3.5.

Figure 26 – Structure et services de la couche de liaison de données (Maître)

La Figure 27 illustre une vue globale de la structure et des services de la couche de liaison de données du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
System Management	Gestion Système
Application Layer	Couche d'Application
On-request Data handler	Module de traitement de Données à la Demande
Process Data handler	Module de traitement des Données de Processus
Line handler	Module de traitement de ligne
Device DL-mode handler	Module de traitement de mode DL du Dispositif
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Inactive	Inactive
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer	Couche physique

Figure 27 – Structure et services de la couche de liaison de données (Dispositif)

7.2 Services de la couche liaison de données

7.2.1 Services de la section B de couche DL

7.2.1.1 Aperçu des services au sein du Maître et du Dispositif

L'Article 7 définit les services de la couche de liaison de données à fournir à la couche d'application et la gestion système via ses interfaces externes. Le Tableau 14 énumère les attributions du Maître et du Dispositif quant à leur rôle d'initiateur ou de destinataire des services individuels de liaison de données (DL). Des champs vides indiquent la non-disponibilité de ce service au niveau du Maître ou du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 14 – Attributions des services au sein du Maître et du Dispositif

Désignation du service	Maître	Dispositif
DL_ReadParam	R	I
DL_WriteParam	R	I
DL_ISDUTransport	R	I
DL_ISDUAbort	R	I
DL_PDOutputUpdate	R	
DL_PDOutputTransport		I
DL_PDInputUpdate		R
DL_PDInputTransport	I	
DL_PDCycle	I	I
DL_SetMode	R	
DL_Mode	I	I
DL_Event	I	R
DL_EventConf	R	
DL_EventTrigger		R
DL_Control	I / R	R / I
DL_Read	R	I
DL_Write	R	I
Légende (voir 3.3.4) I Initiateur du service R Destinataire (Répondeur) du service		

En ce qui concerne les conventions et la manière de lire les descriptifs de services fournis aux 7.2, 8.2, 9.2.2 et 9.3.2, voir 3.3.

7.2.1.2 DL_ReadParam

Le service DL_ReadParam est utilisé par la couche AL pour lire une valeur de paramètre provenant du Dispositif, via le canal de communication de page. Les paramètres des primitives de service sont énumérés dans le Tableau 15.

Tableau 15 – DL_ReadParam

Nom du paramètre	.req	.cnf	.ind
Argument	M		M
Adresse	M		M
Résultat (+)		S	
Valeur		M	
Résultat (-)		S	
ErrorInfo		M	

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Adresse

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Ce paramètre contient l'adresse du paramètre de Dispositif demandé, c'est-à-dire les adresses de paramètres de Dispositif dans le canal de communication de page (voir le Tableau B.1).

Valeurs autorisées: 0 à 31

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Valeur
Ce paramètre contient la valeur du paramètre de Dispositif à lire.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.3 DL_WriteParam

Le service DL_WriteParam est utilisé par la couche AL pour écrire une valeur de paramètre vers le Dispositif via le canal de communication de page. Les paramètres des primitives de service sont énumérés dans le Tableau 16.

Tableau 16 – DL_WriteParam

Nom du paramètre	.req	.cnf	.ind
Argument	M		M
Adresse	M		M
Valeur	M		M
Résultat(+)		S	
Résultat (-)		S	
ErrorInfo		M	

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Adresse
Ce paramètre contient l'adresse du paramètre de Dispositif demandé, c'est-à-dire les adresses de paramètres de Dispositif dans le canal de communication de page.

Valeurs autorisées: 16 à 31, selon les droits d'accès aux paramètres de Dispositif

Valeur
Ce paramètre contient la valeur du paramètre de Dispositif à écrire.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.4 DL_Read

Le service DL_Read est utilisé par la gestion système pour lire une valeur de paramètre de Dispositif via le canal de communication de page. Les paramètres des primitives de service sont énumérés dans le Tableau 17.

Tableau 17 – DL_Read

Nom du paramètre	.req	.cnf	.ind
Argument	M		M
Adresse	M		M
Résultat(+)		S	
Valeur		M	
Résultat (-)		S	
ErrorInfo		M	

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Adresse
Ce paramètre contient l'adresse du paramètre de Dispositif demandé, c'est-à-dire les adresses de paramètres de Dispositif dans le canal de communication de page (voir le Tableau B.1).

Valeurs autorisées: 0 à 15, selon les droits d'accès aux paramètres de Dispositif

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Valeur
Ce paramètre contient les valeurs du paramètre de Dispositif à lire.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.5 DL_Write

Le service DL_Write est utilisé par la gestion système pour écrire, vers le Dispositif, une valeur de paramètre de Dispositif, via le canal de communication de page. Les paramètres des primitives de service sont énumérés dans le Tableau 18.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 18 – DL_Write

Nom du paramètre	.req	.cnf	.ind
Argument	M		M
Adresse	M		M
Valeur	M		M
Résultat(+)		S	
Résultat (-)		S	
ErrorInfo		M	

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Adresse
Ce paramètre contient l'adresse du paramètre de Dispositif demandé, c'est-à-dire les adresses de paramètres de Dispositif dans le canal de communication de page.

Valeurs autorisées: 0 à 15, selon les droits d'accès aux paramètres

Valeur
Ce paramètre contient la valeur du paramètre de Dispositif à écrire.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.6 DL_ISDUTransport

Le service DL_ISDUTransport est utilisé pour transporter une ISDU. Ce service est utilisé par le Maître pour envoyer une demande de service de la couche d'application du Maître vers le Dispositif. Il est utilisé par le Dispositif pour envoyer une réponse de service au Maître depuis la couche d'application du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 19.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 19 – DL_ISDUTransport

Nom du paramètre	.req	.ind	.cnf	.res
Argument	M	M		
ValueList	M	M		
Résultat(+)			S	S
Données			C	C
Qualificatif			M	M
Résultat (-)			S	S
ISDUTransportErrorInfo			M	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

ValueList
Ce paramètre contient les paramètres de fonctionnement pertinents

Type de paramètre: Enregistrement

Index
Valeurs autorisées: 2 à 65 535 (Voir B.2.1 pour les contraintes)

Sous-index
Valeurs autorisées: 0 à 255

Données
Type de paramètre: Chaîne d'octets

Sens
Valeurs autorisées:
READ (opération de lecture),
WRITE (opération d'écriture)

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Données
Type de paramètre: Chaîne d'octets

Qualificatif
Valeurs autorisées: une réponse de service I du Dispositif conformément au Tableau A.12

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ISDUTransportErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant),
ISDU_TIMEOUT (Expiration de la période d'acquittement des ISDU, voir le Tableau 97),
ISDU_NOT_SUPPORTED (ISDU non mise en œuvre),
VALUE_OUT_OF_RANGE (La valeur du paramètre de service ne respecte pas les plages définies)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.1.7 DL_ISDUAbort

Le service DL_ISDUAbort abandonne la transmission ISDU en cours. Ce service n'a pas de paramètres. Les primitives de service sont énumérées dans le Tableau 20.

Tableau 20 – DL_ISDUAbort

Nom du paramètre	.req	.cnf
<Aucun>		

Le service renvoie une confirmation après abandon de la transmission des ISDU.

7.2.1.8 DL_PDOutputUpdate

La couche d'application du Maître utilise le service DL_PDOutputUpdate pour mettre à jour les données de sortie (Données de Processus du Maître vers le Dispositif) sur la couche de liaison de données. Les paramètres des primitives de service sont énumérés dans le Tableau 21.

Tableau 21 – DL_PDOutputUpdate

Nom du paramètre	.req	.cnf
Argument	M	
OutputData	M	
Résultat(+)		S
TransportStatus		M
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

OutputData
Ce paramètre contient les Données de Processus fournies par la couche d'application.

Type de paramètre: Chaîne d'octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

TransportStatus
Ce paramètre indique si la couche de liaison de données est dans un état qui permet le transfert des données au(x) partenaire(s) de communication.

Valeurs autorisées:
OUI (transmission de données autorisée),
NON (transmission de données non autorisée),

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.9 DL_PDOutputTransport

La couche de liaison de données sur le Dispositif utilise les services DL_PDOutputTransport pour transférer le contenu des Données de Processus de sortie vers la couche d'application (du Maître vers le Dispositif). Les paramètres des primitives de service sont énumérés dans le Tableau 22.

Tableau 22 – DL_PDOutputTransport

Nom du paramètre	.ind
Argument	M
OutputData	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

OutputData
Ce paramètre contient les Données de Processus à transmettre à la couche d'application.

Type de paramètre: Chaîne d'octets

7.2.1.10 DL_PDInputUpdate

La couche d'application du Dispositif utilise les services DL_PDInputUpdate pour mettre à jour les données de sortie (Données de Processus du Dispositif vers le Maître) sur la couche de liaison de données. Les paramètres des primitives de service sont énumérés dans le Tableau 23.

Tableau 23 – DL_PDInputUpdate

Nom du paramètre	.req	.cnf
Argument	M	
InputData	M	
Résultat(+)		S
TransportStatus		M
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

InputData
Ce paramètre contient les Données de Processus fournies par la couche d'application.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

TransportStatus

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Ce paramètre indique si la couche de liaison de données est dans un état qui permet le transfert des données au(x) partenaire(s) de communication.

Valeurs autorisées:
OUI (transmission de données autorisée),
NON (transmission de données non autorisée),

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.1.11 DL_PDInputTransport

La couche de liaison de données sur le Maître utilise les services DL_PDInputTransport pour transférer le contenu des données d'entrée (Données de Processus du Dispositif vers le Maître) vers la couche d'application. Les paramètres des primitives de service sont énumérés dans le Tableau 24.

Tableau 24 – DL_PDInputTransport

Nom du paramètre	.ind
Argument	M
InputData	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

InputData
Ce paramètre contient les Données de Processus à transmettre à la couche d'application.

Type de paramètre: Chaîne d'octets

7.2.1.12 DL_PDCycle

La couche de liaison de données utilise le service DL_PDCycle pour indiquer à la couche d'application la fin d'un cycle de Données de Processus. Ce service n'a pas de paramètres. Les primitives de service sont énumérées dans le Tableau 25.

Tableau 25 – DL_PDCycle

Nom du paramètre	.ind
<none>	

7.2.1.13 DL_SetMode

Le service DL_SetMode est utilisé par la gestion système pour configurer les diagrammes d'états de la couche de liaison de données et envoyer les valeurs de caractéristiques de fonctionnement requises à la couche de liaison de données. Les paramètres des primitives de service sont énumérés dans le Tableau 26.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 26 – DL_SetMode

Nom du paramètre	.req	.cnf
Argument	M	
Mode	M	
ValueList	U	
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Mode
Ce paramètre indique le mode demandé de la DL du Maître sur un port individuel.

Valeurs autorisées:
INACTIVE (Le module de traitement doit passer à l'état INACTIVE),
STARTUP (Le module de traitement doit passer à l'état STARTUP),
PREOPERATE (Le module de traitement doit passer à l'état PREOPERATE),
OPERATE (Le module de traitement doit passer à l'état OPERATE)

ValueList
Ce paramètre contient les paramètres de fonctionnement pertinents.

Structure des données: enregistrement

M-sequenceTime: (à propager au module de traitement des messages)

M-sequenceType: (à propager au module de traitement des messages)
Valeurs autorisées:
TYPE_0,

TYPE_1_1, TYPE_1_2, TYPE_1_V,

TYPE_2_1, TYPE_2_2, TYPE_2_3, TYPE_2_4, TYPE_2_5, TYPE_2_6, TYPE_2_V
(Le TYPE_1_1 force le mode d'entrelacement de la transmission de Données de Processus et de Données à la Demande, voir 7.3.4.2)

PDInputLength: (à propager au module de traitement des messages)

PDOutputLength: (à propager au module de traitement des messages)

OnReqDataLengthPerMessage: (à propager au module de traitement des messages)

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (service non disponible en l'état courant),
PARAMETER_CONFLICT (cohérence du jeu de paramètres non respectée)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.1.14 DL_Mode

La couche DL utilise le service DL_Mode pour informer la gestion système qu'un certain état opérationnel a été atteint. Les paramètres des primitives de service sont énumérés dans le Tableau 27.

Tableau 27 – DL_Mode

Nom du paramètre	.ind
Argument	M
RealMode	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

RealMode
Ce paramètre indique l'état du module de traitement DL-mode.

Valeurs autorisées:

INACTIVE	(Le module de traitement est passé à l'état INACTIVE)
COM1	(Mode COM1 établi)
COM2	(Mode COM2 établi)
COM3	(Mode COM3 établi)
COMLOST	(Perte de communication)
ESTABCOM	(Le module de traitement est passé à l'état EstablishCom)
STARTUP	(Le module de traitement est passé à l'état STARTUP)
PREOPERATE	(Le module de traitement est passé à l'état PREOPERATE)
OPERATE	(Le module de traitement est passé à l'état OPERATE)

7.2.1.15 DL_Event

Le service DL_Event indique un état en instance ou des informations d'erreur. La cause d'un Événement se trouve dans un Dispositif et l'application du Dispositif déclenche le transfert de l'Événement. Les paramètres des primitives de service sont énumérés dans le Tableau 28.

Tableau 28 – DL_Event

Nom du paramètre	.req	.ind
Argument	M	M
Instance	M	M
Type	M	M
Mode	M	M
EventCode	M	M
EventsLeft		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Instance
Ce paramètre indique l'origine de l'Événement.

Valeurs autorisées: Application (voir le Tableau A.17)

Type
Ce paramètre indique la catégorie de l'Événement.

Valeurs autorisées: ERROR, WARNING, NOTIFICATION (voir le Tableau A.19)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Mode
Ce paramètre indique le mode de l'Événement.

Valeurs autorisées: SINGLESHOT, APPEARS, DISAPPEARS (voir le Tableau A.20)

EventCode
Ce paramètre contient un code identifiant un certain Événement (voir le Tableau D.1).

Type de paramètre: Entier 16-Bit non signé

EventsLeft
Ce paramètre indique le nombre d'Événements non traités.

7.2.1.16 DL_EventConf

Le service DL_EventConf confirme les Événements transmis via le module de traitement d'Événements. Ce service n'a pas de paramètres. Les primitives de service sont énumérées dans le Tableau 29.

Tableau 29 – DL_EventConf

Nom du paramètre	.req	.cnf
<Aucun>		

7.2.1.17 DL_EventTrigger

La demande DL_EventTrigger démarre la signalisation d'Événements (voir drapeau d'Événement en Figure A.3) et fige la mémoire d'Événements dans la couche DL. La confirmation est renvoyée après traitement des Événements activés. Les demandes DL_EventTrigger supplémentaires sont ignorées jusqu'à ce que la précédente ait été confirmée (voir 7.3.8, 8.3.3 et la Figure 64). Ce service n'a pas de paramètres. Les primitives de service sont énumérées dans le Tableau 30.

Tableau 30 – DL_EventTrigger

Nom du paramètre	.req	.cnf
<Aucun>		

7.2.1.18 DL_Control

Le Maître utilise le service DL_Control pour acheminer des informations de commande, via le mécanisme MasterCommand, à l'application spécifique à la technologie du Dispositif correspondant et pour obtenir des informations de commande via le mécanisme indicateur (drapeau) d'état des PD (voir A.1.5) et le service PDInStatus (voir 7.2.2.5). Les paramètres des primitives de service sont énumérés dans le Tableau 31.

Tableau 31 – DL_Control

Nom du paramètre	.req	.ind
Argument	M	M
ControlCode	M	M(=)

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ControlCode
Ce paramètre indique l'état du qualificatif des Données de Processus (PD)

Valeurs autorisées:

VALID	(Données de Processus d'entrée valides; voir 7.2.2.5, 8.2.2.12)
INVALID	(Données de Processus d'entrée non valides)
PDOUTVALID	(Données de Processus de sortie valides; voir 7.3.7.1)
PDOUTINVALID	(Données de Processus de sortie non valides ou manquantes)

7.2.2 Services de la section A de couche DL

7.2.2.1 Aperçu

Comme défini en 7.1, la couche de liaison de données est divisée en une section supérieure B (DL-B) et une section inférieure A (DL-A). La couche DL-A comprend le module de traitement des messages, comme illustré en Figure 26 et en Figure 27.

Le module de traitement des messages du Maître code les commandes et données en messages et les envoie au Dispositif connecté via la couche physique. Il reçoit des messages du Dispositif via la couche physique et achemine leur contenu aux modules de traitement correspondants sous la forme d'une confirmation. Lorsque le "drapeau d'Événement" est établi dans un message de Dispositif (voir A.1.5), le module de traitement des messages du Maître invoque un service EventFlag pour solliciter le module de traitement d'Événements.

Le module de traitement des messages du Maître doit utiliser une stratégie de renouvellement de tentative en cas de message altéré, c'est-à-dire s'il reçoit d'un Dispositif une somme de contrôle incorrecte ou en l'absence de toute somme de contrôle. Dans ces cas-là, le Maître doit répéter le message deux fois (voir le Tableau 97). En cas d'échec des nouvelles tentatives, une confirmation négative doit être fournie et le Maître doit relancer la communication via le module de traitement de Port-x, en commençant par une réactivation.

Au bout d'une phase de démarrage, le module de traitement des messages lance un cycle opérationnel, le type de séquence M et la durée de cycle étant fournis par le service DL_SetMode.

Le Tableau 32 énumère les attributions du Maître et du Dispositif quant à leurs rôles d'initiateur (I) ou de destinataire (R) dans le contexte de l'exécution de leurs services DL-A individuels.

Tableau 32 – Services DL-A au sein du Maître et du Dispositif

Désignation du service	Maître	Dispositif
OD	R	I
PD	R	I
EventFlag	I	R
PDInStatus	I	R
MHInfo	I	I
ODTrig	I	
PDTrig	I	

7.2.2.2 OD

Le service OD est utilisé pour configurer les Données à la Demande du prochain message à envoyer. La confirmation du service contient les données en provenance du récepteur. Les paramètres des primitives de service sont énumérés dans le Tableau 33.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 33 – OD

Nom du paramètre	.req	.ind	.rsp	.cnf
Argument	M	M		
RWDirection	M	M		
ComChannel	M	M		
AddressCtrl	M	M		
Longueur	M	M		
Données	C	C		
Résultat(+)			S	S
Données			C	C(=)
Longueur			M	M
Résultat (-)			S	S
ErrorInfo			M	M(=)

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

RWDirection
Ce paramètre indique le sens, lecture ou écriture.

Valeurs autorisées:
READ (opération de lecture),
WRITE (opération d'écriture)

ComChannel
Ce paramètre indique le canal de communication sélectionné pour la transmission.

Valeurs autorisées: DIAGNOSIS, PAGE, ISDU (voir le Tableau A.1)

AddressCtrl
Ce paramètre contient l'adresse ou la valeur du contrôle de flux (voir A.1.2).

Valeurs autorisées: 0 à 31

Longueur
Ce paramètre contient la longueur des données à transmettre.

Valeurs autorisées: 0 à 32

Données
Ce paramètre contient les données à transmettre.

Type de Données: Chaîne d'octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Données
Ce paramètre contient la valeur des données à lire.

Longueur
Ce paramètre contient la longueur du paquet de données reçu.

Valeurs autorisées: 0 à 32

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.2.3 PD

Le service PD est utilisé pour configurer les Données de Processus à envoyer par l'intermédiaire du canal de communication du processus. La confirmation du service contient les données en provenance du récepteur. Les paramètres des primitives de service sont énumérés dans le Tableau 34.

Tableau 34 – PD

Nom du paramètre	.req	.ind	.rsp	.cnf
Argument	M	M		
PDInAddress	C	C(=)		
PDInLength	C	C(=)		
PDout	C	C(=)		
PDOutAddress	C	C(=)		
PDOutLength	C	C(=)		
Résultat(+)			S	S
PDin			C	C(=)
Résultat (-)			S	S
ErrorInfo			M	M(=)

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

PDInAddress
Ce paramètre contient l'adresse des Données de Processus d'entrée demandées (voir 7.3.4.2).

PDInLength
Ce paramètre contient la longueur des Données de Processus d'entrée demandées.

Valeurs autorisées: 0 à 32

PDOut
Ce paramètre contient les Données de Processus à transférer du Maître vers le Dispositif.

Type de Données: Chaîne d'octets

PDOutAddress
Ce paramètre contient l'adresse des Données de Processus de sortie transmises (voir 7.3.4.2).

PDOutLength
Ce paramètre contient la longueur des Données de Processus de sortie transmises.

Valeurs autorisées: 0 à 32

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Résultat(+)

Ce paramètre de sélection indique que le service a été exécuté avec succès.

PDin
Ce paramètre contient les Données de Processus à transférer du Dispositif vers le Maître.

Type de Données: Chaîne d'octets

Résultat (-)

Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_COMM (pas de communication disponible),
STATE_CONFLICT (service non disponible en l'état courant)

7.2.2.4 EventFlag

Le service EventFlag établit ou notifie l'état du "drapeau d'Événement " (voir A.1.5) lors d'une communication cyclique. Les paramètres des primitives de service sont énumérés dans le Tableau 35.

Tableau 35 – EventFlag

Nom du paramètre	.ind	.req
Argument		
Flag (indicateur)	M	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Flag
Ce paramètre contient la valeur du "drapeau d'Événement ".

Valeurs autorisées:
TRUE ("Drapeau d'Événement"= 1)
FALSE ("Drapeau d'Événement" = 0)

7.2.2.5 PDInStatus

Le service PDInStatus établit et notifie le qualificatif de validité des Données de Processus d'entrée. Les paramètres des primitives de service sont énumérés dans le Tableau 36.

Tableau 36 – PDInStatus

Nom du paramètre	.req	.ind
Argument		
État	M	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

État

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Ce paramètre contient l'indication de validité des Données de Processus d'entrée transmises.

Valeurs autorisées:
VALID (Données de Processus d'entrée valides sur la base du drapeau indicateur d'état des PD (voir A.1.5); voir 7.2.1.18)
INVALID (Données de Processus d'entrée non valides)

7.2.2.6 MHInfo

Le service MHInfo notifie un fonctionnement exceptionnel dans le module de traitement des messages. Les paramètres du service sont énumérés dans le Tableau 37.

Tableau 37 – MHInfo

Nom du paramètre	.ind
Argument	
MHInfo	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

MHInfo
Ce paramètre contient l'indication d'une situation exceptionnelle du module de traitement des messages.

Valeurs autorisées:
COMLOST (perte de communication),
ILLEGAL_MESSAGETYPE (détection d'un type de séquence M imprévu)
CHECKSUM_MISMATCH (détection d'une erreur de somme de contrôle)

7.2.2.7 ODTrig

Le service ODTrig est disponible uniquement au niveau du Maître. Le service déclenche le module de traitement de Données à la Demande ainsi que les modules de traitement des ISDU, des Commandes ou d'Événements actuellement chargés de fournir les Données à la Demande (via le service OD) pour le prochain message de Maître. Les paramètres du service sont énumérés dans le Tableau 38.

Tableau 38 – ODTrig

Nom du paramètre	.ind
Argument	
DataLength	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

DataLength
Ce paramètre contient l'espace disponible, par message, pour les Données à la Demande (OD).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.2.2.8 PDTrig

Le service PDTrig est disponible uniquement au niveau du Maître. Le service déclenche le module de traitement des Données de Processus qui fournit les Données de Processus (PD) pour le prochain message de Maître.

Les paramètres du service sont énumérés dans le Tableau 39.

Tableau 39 – PDTrig

Nom du paramètre	.ind
Argument	
DataLength	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

DataLength
Ce paramètre contient l'espace disponible, par message, pour les Données de Processus (PD).

7.3 Protocole de la couche liaison de données

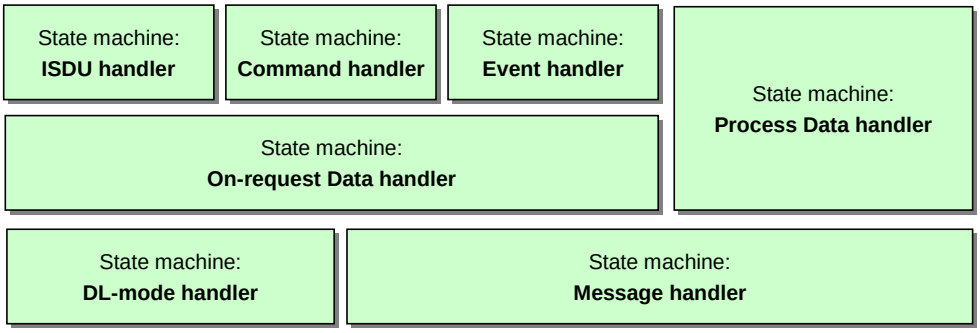
7.3.1 Aperçu

La Figure 26 et la Figure 27 illustrent la structure de la couche de liaison de données et de ses constituants; un module de traitement de mode DL, un module de traitement des messages, un module de traitement des Données de Processus, et un module de traitement de Données à la Demande permettant d'assurer les services spécifiés. Dans les paragraphes 7.3.2 à 7.3.8, le comportement (dynamique) de ces modules de traitement est défini par des diagrammes d'états UML et des tableaux de transitions.

Le module de traitement de Données à la Demande prend en charge trois types indépendants de données: ISDU, commande et Événement. De ce fait, trois diagrammes d'états supplémentaires collaborent avec le diagramme d'états du module de traitement de Données à la Demande, comme illustré en Figure 28. Des diagrammes de séquence ou d'activité complémentaires présentent certains cas d'usage spécifiques. Voir CEI/TR 62390 et ISO/CEI 19505.

Les éléments couverts par chaque module de traitement, tels que les messages, les procédures de réactivation, le mode d'entrelacement, les ISDU (Unités de données de service indexées) et les Événements sont définis dans le contexte du module de traitement correspondant.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
State machine:	Diagramme d'états:
ISDU handler	Module de traitement des ISDU
Command handler	Module de traitement des commandes
Event handler	Module de traitement d'Événements
On-request Data handler	Module de traitement de Données à la Demande
Process Data handler	Module de traitement des Données de Processus
DL-mode handler	Module de traitement de mode DL
Message handler	Module de traitement des messages

Figure 28 – Diagrammes d'états de la couche de liaison de données

7.3.2 Module de traitement de mode DL

7.3.2.1 Généralités

Le module de traitement de mode DL du Maître, illustré en Figure 26 est chargé de la communication SDCI au moyen des services de la Couche Physique (PL) et des invocations réglementaires internes pour contrôler et surveiller le module de traitement des messages ainsi que les états des autres modules de traitement.

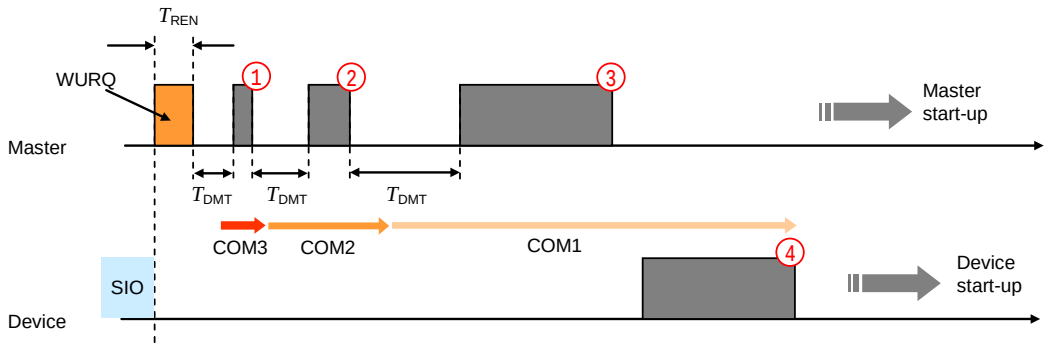
Le module de traitement de mode DL du Dispositif illustré en Figure 27 est chargé de la détection d'une demande de réactivation et de l'établissement de la communication. Il reçoit des MasterCommands pour se synchroniser avec les états du module de traitement de mode DL du Maître STARTUP, PREOPERATE et OPERATE et gère l'activation et la désactivation des modules de traitement en fonction des besoins.

7.3.2.2 Procédures de réactivation et règles de conformité du Dispositif

La gestion système déclenche les actions suivantes sur la couche de liaison de données, à l'aide du service DL_SetMode (mode demandé = STARTUP).

Le module de traitement de mode DL du Maître tente d'établir la communication via une demande de réactivation (PL_WakeUp.req) suivie par un message d'essai avec une séquence M de TYPE_0 (lecture de "MinCycleTime") conformément à la séquence illustrée en Figure 29.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master	Maître
Device	Dispositif
Master start-up	Démarrage du Maître
Device start-up	Démarrage du Dispositif

Figure 29 – Exemple d'une tentative d'établissement de communication

Après la demande de réactivation (WURQ), spécifiée en 5.3.3.3, le module de traitement de mode DL demande au module de traitement des messages d'envoyer le premier message d'essai au bout d'une période T_{REN} (voir le Tableau 9) et T_{DMT} (voir le Tableau 40). Les vitesses de transmission spécifiées COM1, COM2 et COM3 sont utilisées par ordre décroissant jusqu'à ce qu'une réponse soit obtenue, comme illustré dans l'exemple de la Figure 29:

Étape ① Message de Maître à une vitesse de transmission de COM3 (voir le Tableau 8).

Étape ② Message de Maître à une vitesse de transmission de COM2 (voir le Tableau 8).

Étape ③ Message de Maître à une vitesse de transmission de COM1 (voir le Tableau 8).

Étape ④ Message de réponse du Dispositif à une vitesse de transmission de COM1.

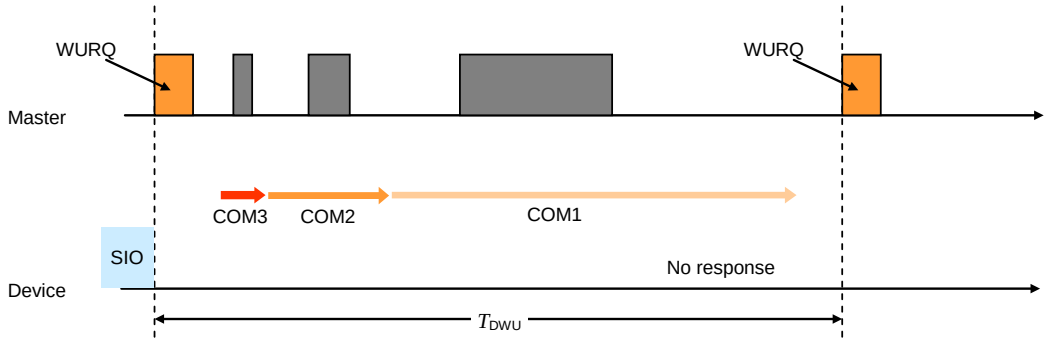
Avant de lancer un (nouveau) message, le module de traitement de mode DL doit attendre pendant une période d'au moins T_{DMT}. T_{DMT} est spécifiée dans le Tableau 40.

En ce qui concerne la prise en charge des vitesses de transmission, la règle de conformité suivante s'applique aux Dispositifs:

- un Dispositif doit prendre en charge uniquement l'une des vitesses de transmission de COM1, COM2 ou COM3.

En cas d'échec d'une tentative d'établissement de communication, le module de traitement de mode DL du Maître ne doit lancer une nouvelle tentative (procédure de réactivation) qu'après un délai T_{DWU}, comme illustré en Figure 30 et spécifié dans le Tableau 40.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

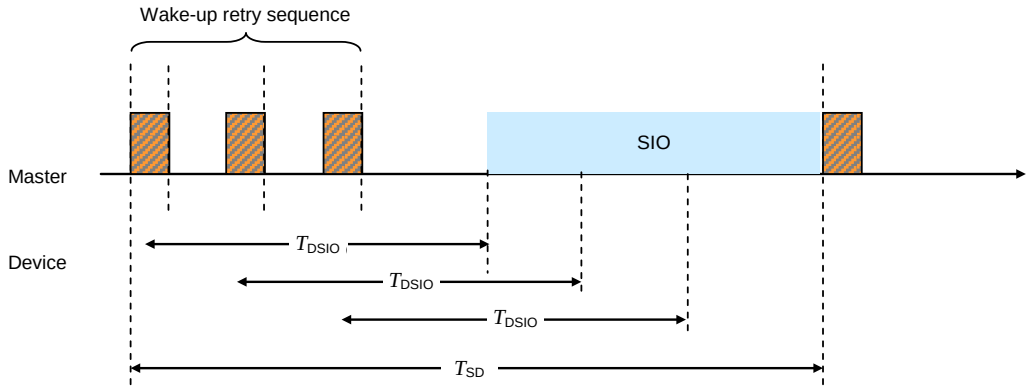


Légende

Anglais	Français
Master	Maître
Device	Dispositif
No response	Pas de réponse

Figure 30 – Échec d'une tentative d'établissement de communication

Le Maître doit effectuer jusqu'à n_{WU}+1 demandes de réactivation successives, comme illustré en Figure 31. Si cette séquence initiale de nouvelle tentative de réactivation échoue, le Dispositif doit réinitialiser sa ligne C/Q en mode SIO après une période T_{DSIO} (T_{DSIO} est redéclenchée dans le Dispositif après chaque détection de WURQ). Le Maître ne doit déclencher une nouvelle séquence de tentative de réactivation qu'au bout d'un délai T_{SD}.



Légende

Anglais	Français
Master	Maître
Device	Dispositif
Wake-up retry sequence	Séquence de nouvelle tentative de réactivation

Figure 31 – Stratégie de nouvelle tentative d'établissement de communication

La couche DL du Maître doit demander à la PL de passer en mode SIO après échec d'une séquence de nouvelle tentative de réactivation.

Les valeurs de temporisation de la procédure et des nouvelles tentatives de réactivation sont spécifiées dans le Tableau 9 ainsi que dans le Tableau 40. Elles sont définies du point de vue du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 40 – Caractéristiques des procédures et des nouvelles tentatives de réactivation

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
T_{DMT}	Délai du message du Maître	27	n/a	37	T_{BIT}	Temps binaire de la vitesse de transmission de données ultérieure
T_{DSIO}	Délai E/S normalisé	60	n/a	300	ms	Au bout de T_{DSIO}, le dispositif se replie en mode SIO (s'il est pris en charge)
T_{DWU}	Délai d'une nouvelle tentative de réactivation	30	n/a	50	ms	Au bout de T_{DWU}, le Maître répète la demande de réactivation
n_{WU}	Nombre de nouvelles tentatives de réactivation	2	2	2		Nombre de nouvelles tentatives de demande de réactivation
T_{SD}	Temps de détection du Dispositif	0,5	n/a	1	s	Temps entre 2 séquences de demande de réactivation. (Voir NOTE)
NOTE Caractéristique du Maître.						

La couche de liaison de données du Maître doit arrêter la procédure d'établissement de la communication lorsqu'elle trouve un Dispositif communiquant et doit rendre compte du mode COMx détecté à la gestion système au moyen d'une indication DL_Mode. Si la procédure échoue, l'erreur correspondante est notifiée en utilisant le même service.

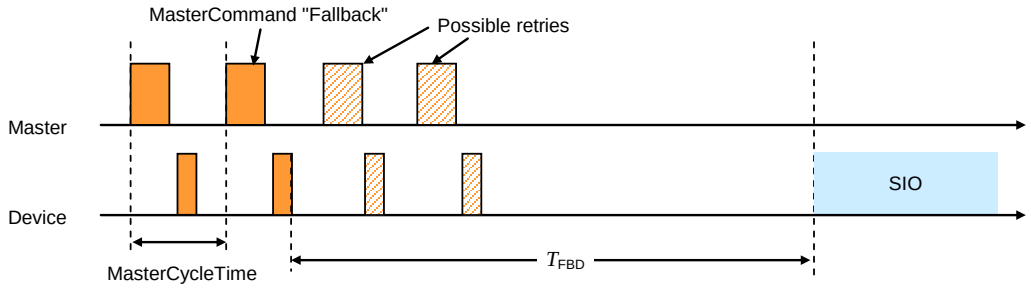
7.3.2.3 Procédure de repli

La gestion système induit les actions suivantes sur la couche de liaison de données, à l'aide du service DL_SetMode (mode = INACTIVE):

- Une MasterCommand "Repli" (voir le Tableau B.2) force le Dispositif à passer en mode SIO.
- Le Dispositif doit effectuer la transition vers le mode SIO après 3 durées MasterCycleTimes et/ou dans les 500 ms après la commande MasterCommand "Repli". Ceci tient compte des éventuelles tentatives en cas d'échec de la MasterCommand, indiquée par une réponse négative du Dispositif.

La Figure 32 illustre la procédure de repli ainsi que les contraintes liées aux nouvelles tentatives et à la temporisation.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
MasterCommand "Fallback"	MasterCommand "Repli"
Master	Maître
Device	Dispositif
Possible retries	Nouvelles tentatives possibles.

Figure 32 – Procédure de repli

Le Tableau 41 définit les caractéristiques temporelles du repli. Une description plus détaillée est fournie en A.2.6.

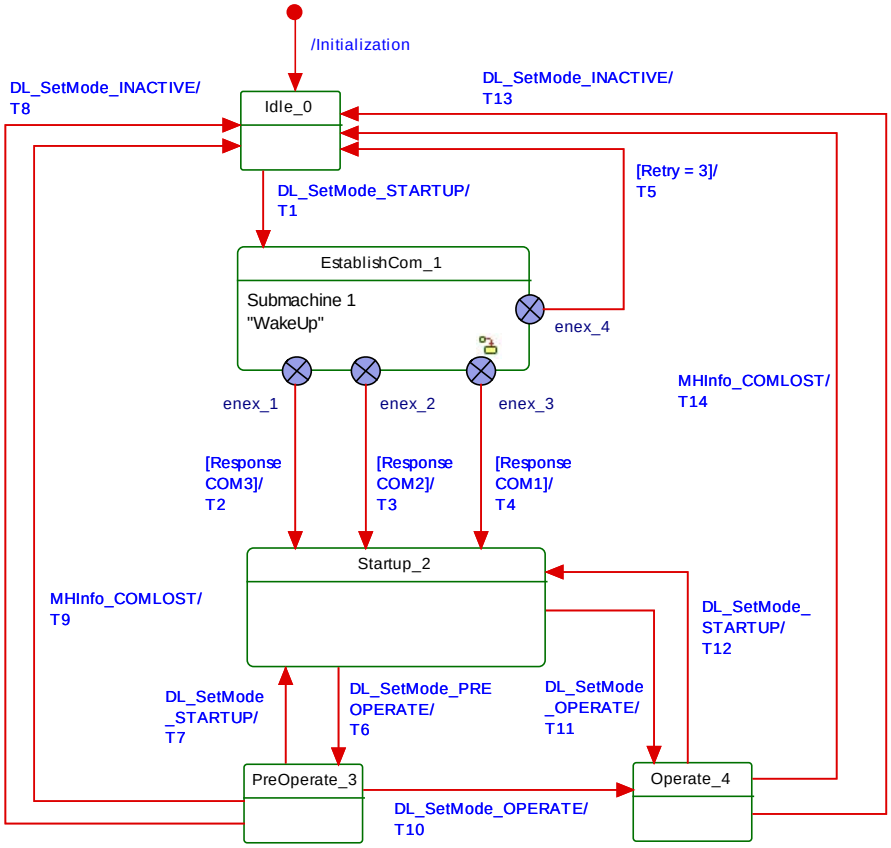
Tableau 41 – Caractéristiques temporelles du repli

Propriété	Désignation	Minimale	Type	Maximale	Unité	Remarque
T_{FBD}	Délai de repli	3 MasterCycle-Times (OPERATE) ou 3 $T_{initcyc}$ (PREOPERATE)	n/a	500	ms	Au bout d'un temps T_{FBD} le Dispositif doit passer en mode SIO (voir la Figure 32)

7.3.2.4 Diagramme d'états du module de traitement de mode DL du Maître

La Figure 33 illustre le diagramme d'états du module de traitement de mode DL du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation
Retry	Nouvelle tentative
Submachine 1 "WakeUp"	Sous-machine 1 "Réactivation"
Response	Réponse

Figure 33 – Diagramme d'états du module de traitement de mode DL du Maître

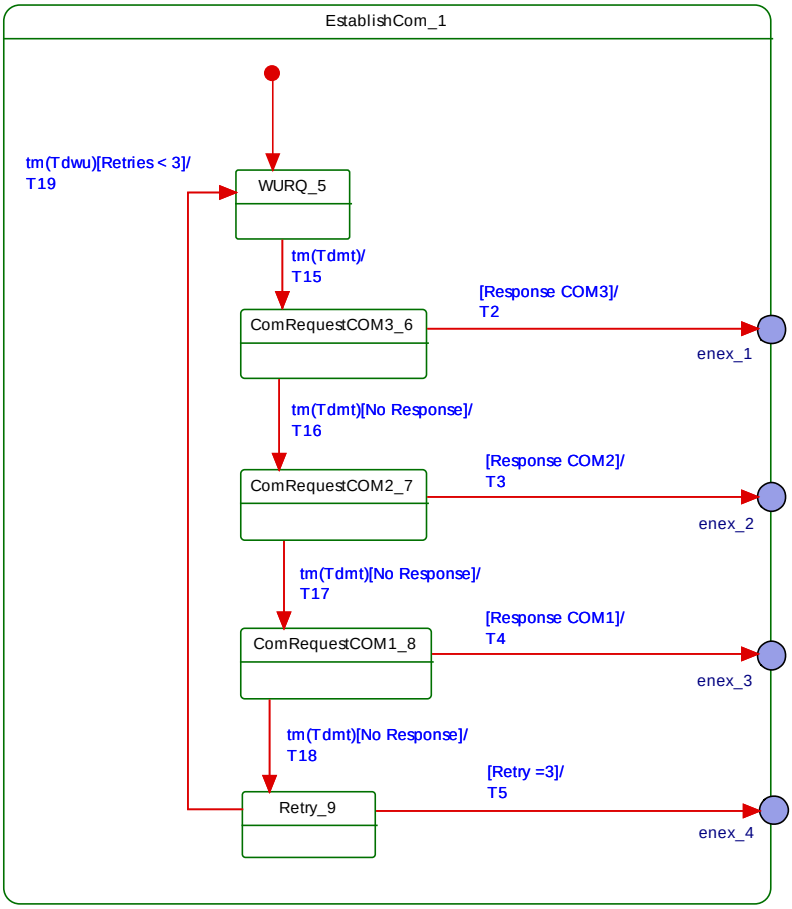
NOTE Les conventions utilisées dans les types de diagrammes UML sont définies en 3.3.7.

Après réception du service DL_SetMode_STARTUP de la Gestion Système, le module de traitement de mode DL doit d'abord générer une impulsion de courant de réactivation via le service PL_WakeUp puis établir la communication. Cette procédure est spécifiée dans la sous-machine 1 en Figure 34.

L'état "Startup_2" a pour but de vérifier l'identité d'un Dispositif en utilisant les données de la page de Paramètres Directs (voir la Figure 5). Lorsqu'il est dans l'état "PreOperate_3", le Maître attribue des paramètres au Dispositif en utilisant des ISDU. L'échange cyclique de Données de Processus est réalisé au cours de l'état "Operate". Dans cet état, des Données à la demande supplémentaires, telles que des ISDU, des commandes et des Événements peuvent être transmises en utilisant des types de séquences M appropriés (voir la Figure 37).

Dans les états PreOperate_3 et Operate_4, différents ensembles de modules de traitement sont activés dans le Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Retry	Nouvelle tentative
Response	Réponse
No Response	Pas de réponse

Figure 34 – Sous-machine 1 d'établissement de la communication

Le Tableau 42 présente les tables de transitions d'états du module de traitement de mode DL du Maître.

Tableau 42 – Tables de transitions d'états du module de traitement de mode DL du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Idle_0	Attente d'une demande de réactivation émise par la Gestion Système (SM): DL_SetMode (STARTUP)
EstablishComm_1	Exécution de la procédure de réactivation (sous-machine 1)
Startup_2	La Gestion Système utilise l'état STARTUP pour l'identification du Dispositif, la vérification et la configuration de la communication (voir la Figure 69)
Preoperate_3	Échange de Données à la Demande (paramètre, commandes, Événements) sans Données de Processus
Operate_4	Échange de Données de Processus et de Données à la Demande (paramètre, commandes, Événements)
SM: WURQ_5	Génération de l'impulsion de courant de réactivation: Invocation du service PL-Wake-Up (voir la Figure 11 et 5.3.3.3) et attente de T_{DMT} (voir le Tableau 40).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
SM: ComRequestCOM3_6		Tentative d'envoi du message d'essai à une vitesse de transmission de COM3 via le module de traitement des messages: Invocation de MH_Conf_COMx (voir la Figure 38) et attente de T_{DMT} (voir le Tableau 40).	
SM: ComRequestCOM2_7		Tentative d'envoi du message d'essai à une vitesse de transmission de COM2 via le module de traitement des messages: Invocation de MH_Conf_COMx (voir la Figure 38) et attente de T_{DMT} (voir le Tableau 40).	
SM: ComRequestCOM1_8		Tentative d'envoi du message d'essai à une vitesse de transmission de COM1 via le module de traitement des messages: Invocation de MH_Conf_COMx (voir la Figure 38) et attente de T_{DMT} (voir le Tableau 40).	
SM: Retry_9		Vérification du nombre de nouvelles tentatives	
TRANSITION	**ÉTAT DE LA SOURCE**	**ÉTAT DE LA CIBLE**	**ACTION**
T1	0	1	Set Retry = 0.
T2	1	2	Vitesse de transmission COM3 réussie. Module de traitement des messages activé et configuré en COM3 (voir la Figure 38, Transition T2). Activation du module de traitement des commandes (invocation de CH_Conf_ACTIVE en Figure 51). Renvoi de DL_Mode.ind (STARTUP) et de DL_Mode.ind (COM3) à la SM.
T3	1	2	Vitesse de transmission COM2 réussie. Module de traitement des messages activé et configuré en COM2 (voir la Figure 38, Transition T2). Activation du module de traitement des commandes (invocation de CH_Conf_ACTIVE en Figure 51). Renvoi de DL_Mode.ind (STARTUP) et de DL_Mode.ind (COM2) à la SM.
T4	1	2	Vitesse de transmission COM1 réussie. Module de traitement des messages activé et configuré en COM1 (voir la Figure 38, Transition T2). Activation du module de traitement des commandes (invocation de CH_Conf_ACTIVE en Figure 51). Renvoi de DL_Mode.ind (STARTUP) et de DL_Mode.ind (COM1) à la SM.
T5	1	0	Renvoi de DL_Mode.ind (INACTIF) à la SM.
T6	2	3	La SM a demandé l'état PREOPERATE. Activation du module de traitement de Données à la Demande (invocation de OH_Conf_ACTIVE en Figure 46), du module de traitement des ISDU (invocation de IH_Conf_ACTIVE en Figure 49), et du module de traitement d'Événements (invocation de EH_Conf_ACTIVE en Figure 53). Passage de l'état du module de traitement des messages à PREOPERATE (invocation de MH_Conf_PREOPERATE en Figure 38). Renvoi de DL_Mode.ind (PREOPERATE) à la SM.
T7	3	2	La SM a demandé l'état STARTUP. Passage de l'état du module de traitement des messages à STARTUP (invocation de MH_Conf_STARTUP en Figure 38). Désactivation du module de traitement des Données à la Demande (invocation de OH_Conf_INACTIVE en Figure 46), du module de traitement des ISDU (invocation de IH_Conf_INACTIVE en Figure 49), du module de traitement de commandes (invocation de CH_Conf_INACTIVE en Figure 51) et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 53). Renvoi de DL_Mode.ind (STARTUP) à la SM.
T8	3	0	La SM a demandé le mode SIO. Désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Renvoi de DL_Mode.ind (INACTIF) à la SM.
T9	3	0	Le module de traitement des messages notifie la perte de communication via le service de DL-A MHInfo (COMLOST). Désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Renvoi de DL_Mode.ind (COMLOST) à la SM.
T10	3	4	La SM a demandé l'état OPERATE. Activation du module de traitement des Données de Processus (invocation de PD_Conf_SINGLE si le type de séquence M = TYPE_2_x, ou PD_Conf_INTERLEAVE si le type de séquence M = TYPE_1_1 en Figure 44). Passage de l'état du module de traitement des messages à OPERATE (invocation de MH_Conf_OPERATE en Figure 38). Renvoi de DL_Mode.ind (OPERATE) à la SM.
T11	2	4	La SM a demandé l'état OPERATE. Activation du module de traitement des Données de Processus (invocation de PD_Conf_SINGLE ou PD_Conf_INTERLEAVE en Figure 44, selon la configuration du port du

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

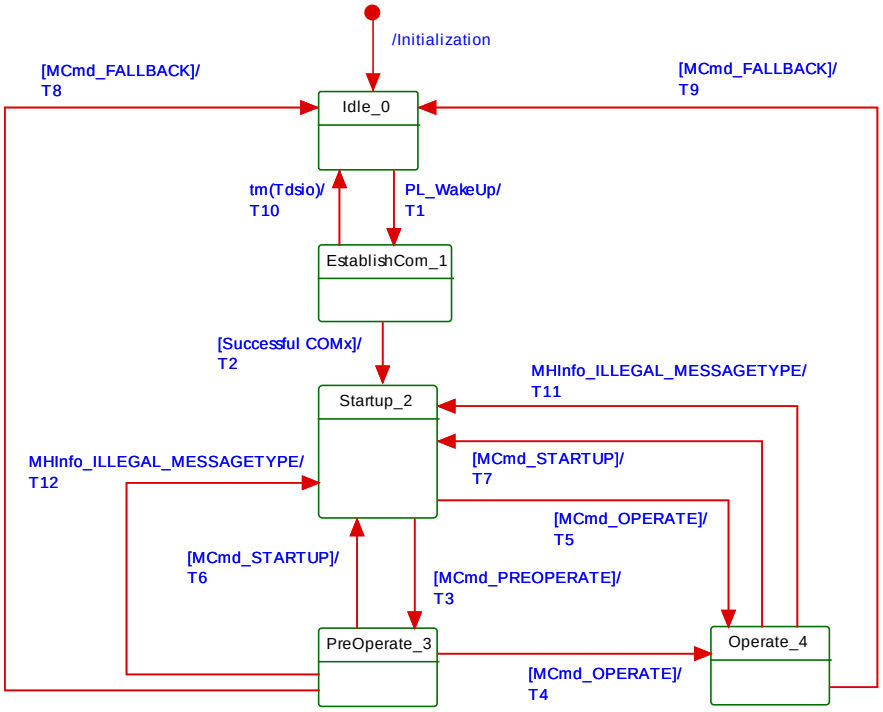
DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
			Maître). Activation des Données à la Demande (invocation de OH_Conf_ACTIVE en Figure 46), ISDU (invocation de IH_Conf_ACTIVE en Figure 49), et du module de traitement d'Événements (invocation de EH_Conf_ACTIVE en Figure 53). Passage de l'état du module de traitement des messages à OPERATE (invocation de MH_Conf_OPERATE en Figure 38). Renvoi de DL_Mode.ind (OPERATE) à la SM.
T12	4	2	La SM a demandé l'état STARTUP. Passage de l'état du module de traitement des messages à STARTUP (invocation de MH_Conf_STARTUP en Figure 38). Désactivation des Données de Processus (invocation de PD_Conf_INACTIVE en Figure 44), des Données à la Demande (invocation de OH_Conf_INACTIVE en Figure 46), des ISDU (invocation de IH_Conf_INACTIVE en Figure 49), et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 53). Renvoi de DL_Mode.ind (STARTUP) à la SM.
T13	4	0	La SM a demandé l'état SIO. Désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Renvoi de DL_Mode.ind (INACTIF) à la SM.
T14	4	0	Le module de traitement des messages notifie la perte de communication via le service de DL-A MHInfo (COMLOST). Désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Renvoi de DL_Mode.ind (COMLOST) à la SM.
T15	5	6	Établissement de la vitesse de transmission au mode COM3.
T16	6	7	Établissement de la vitesse de transmission au mode COM2.
T17	7	8	Établissement de la vitesse de transmission au mode COM1.
T18	8	9	Incrémentation du nombre de Nouvelles tentatives
T19	9	5	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
MH_Conf_COMx	Invocation	Cette invocation force le module de traitement des messages à envoyer un message à la vitesse de transmission COMx demandée et avec une séquence M de TYPE_0 (voir le Tableau 44).
MH_Conf_STARTUP	Invocation	Cette invocation force le module de traitement des messages à passer à l'état STARTUP (voir la Figure 38)
MH_Conf_PREOPERATE	Invocation	Cette invocation force le module de traitement des messages à passer à l'état PREOPERATE (voir la Figure 38)
MH_Conf_OPERATE	Invocation	Cette invocation force le module de traitement des messages à passer à l'état OPERATE (voir la Figure 38)
xx_Conf_ACTIVE	Invocation	Cette invocation active le module de traitement correspondant, où xx remplace MH (module de traitement des messages), OH (module de traitement de Données à la Demande), IH (module de traitement des ISDU), CH (module de traitement des commandes) et/ou EH (module de traitement d'Événements)
xx_Conf_INACTIVE	Invocation	Cette invocation désactive le module de traitement des messages, où xx remplace MH (module de traitement des messages), OH (module de traitement de Données à la Demande), IH (module de traitement des ISDU), CH (module de traitement des commandes) et/ou EH (module de traitement d'Événements)
Retry (Nouvelle tentative)	Variable	Nombre de nouvelles tentatives pour établir la communication

7.3.2.5 Diagramme d'états du module de traitement de mode DL du Dispositif

La Figure 35 illustre le diagramme d'états du module de traitement de mode DL du Dispositif. Dans les états PreOperate_3 et Operate_4, différents ensembles de module de traitement sont activés dans le Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation
Successful COMx	Succès de COMx

Figure 35 – Diagramme d'états du module de traitement de mode DL du Dispositif

Le Maître utilise des MasterCommands (voir le Tableau 42) pour modifier les états du Dispositif en SIO, STARTUP, PREOPERATE et OPERATE. Lorsque le module de traitement des messages détecte des types de séquence M illicites (imprévues), il force le module de traitement de mode DL à passer à l'état STARTUP et à indiquer cet état à sa gestion système (voir 9.3.3.2) aux fins de synchronisation du Maître et du Dispositif.

Le Tableau 43 présente les tables de transitions d'états du module de traitement de mode DL du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 43 – Tables de transitions d'états du module de traitement de mode DL du Dispositif

DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
Idle_0		Attente de détection d'une impulsion de courant de réactivation (PL_WakeUp.ind).	
EstablishComm_1		Module de traitement des messages activé et attente de messages d'essai COMx (voir le Tableau 42)	
Startup_2		Vérification de compatibilité (voir 9.2.3.3)	
Preoperate_3		Échange de Données à la Demande (paramètre, commandes, Événements) sans Données de Processus	
Operate_4		Échange de Données de Processus (PD) et de Données à la Demande (paramètre, commandes, Événements)	
TRANSITION	**ÉTAT DE LA SOURCE**	**ÉTAT DE LA CIBLE**	**ACTION**
T1	0	1	Impulsion de courant de réactivation détectée. Activation du module de traitement des messages (invocation de MH_Conf_ACTIVE en Figure 42). Indication de l'état, via le service DL_Mode.ind (ESTABCOM), à la SM.
T2	1	2	Une des trois vitesses de transmission (COM3, COM2 ou COM1) est établie. Activation du module de traitement de Données à la Demande (invocation de OH_Conf_ACTIVE en Figure 47) et du module de traitement de commandes (invocation de CH_Conf_ACTIVE en Figure 52). Indication de l'état, via le service DL_Mode.ind (COM1, COM2 ou COM3), à la SM.
T3	2	3	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_PREOPERATE). Activation du module de traitement des ISDU (invocation de IH_Conf_ACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_ACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (PREOPERATE), à la SM.
T4	3	4	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_OPERATE). Activation du module de traitement des Données de Processus (invocation de PD_Conf_ACTIVE en Figure 45). Indication de l'état, via le service DL_Mode.ind (OPERATE), à la SM.
T5	2	4	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_OPERATE). Activation du module de traitement des Données de Processus (invocation de PD_Conf_ACTIVE en Figure 45), du module de traitement des ISDU (invocation de IH_Conf_ACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_ACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (OPERATE), à la SM.
T6	3	2	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_STARTUP). Désactivation du module de traitement des ISDU (invocation de IH_Conf_INACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (STARTUP), à la SM.
T7	4	2	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_STARTUP). Désactivation du module de traitement des Données de Processus (invocation de PD_Conf_INACTIVE en Figure 45), du module de traitement des ISDU (invocation de IH_Conf_INACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (STARTUP), à la SM.
T8	3	0	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_FALLBACK). Attente de l'expiration de T_{FBD} puis désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Indication de l'état, via le service DL_Mode.ind (INACTIVE) à la SM (voir la Figure 79 et le Tableau 93).
T9	4	0	Le module de traitement de commandes de dispositif a reçu du Maître la commande MasterCommand (MCmd_FALLBACK). Attente de l'expiration de T_{FBD} puis désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Indication de l'état, via le service DL_Mode.ind

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
			(INACTIVE) à la SM (voir la Figure 79 et le Tableau 93).
T10	1	0	Suite à des échecs de procédures de réactivation (voir la Figure 30), le Dispositif établit le mode SIO configuré, à l'expiration d'un délai T_{DSIO} (voir la Figure 31). Désactivation de tous les modules de traitement (Invocation de xx_Conf_INACTIVE). Indication de l'état, via le service DL_Mode.ind (INACTIVE), à la SM.
T11	4	2	Le module de traitement des messages a détecté un type de séquence M illicite. Désactivation du module de traitement des Données de Processus (invocation de PD_Conf_INACTIVE en Figure 45), du module de traitement des ISDU (invocation de IH_Conf_INACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (STARTUP), à la SM (voir la Figure 79 et le Tableau 93).
T12	3	2	Le module de traitement des messages a détecté un type de séquence M illicite. Désactivation du module de traitement des ISDU (invocation de IH_Conf_INACTIVE en Figure 50) et du module de traitement d'Événements (invocation de EH_Conf_INACTIVE en Figure 54). Indication de l'état, via le service DL_Mode.ind (STARTUP) à la SM (voir la Figure 79 et le Tableau 93).

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
T_{FBD}	Temps	Voir le Tableau 41.
T_{DSIO}	Temps	Voir la Figure 31
MCmd_XXXXXXX	Invocation	Toute MasterCommand reçue par le module de traitement des commandes du Dispositif (voir le Tableau 42 et la Figure 52, état "CommandHandler_2")

7.3.3 Module de traitement des messages

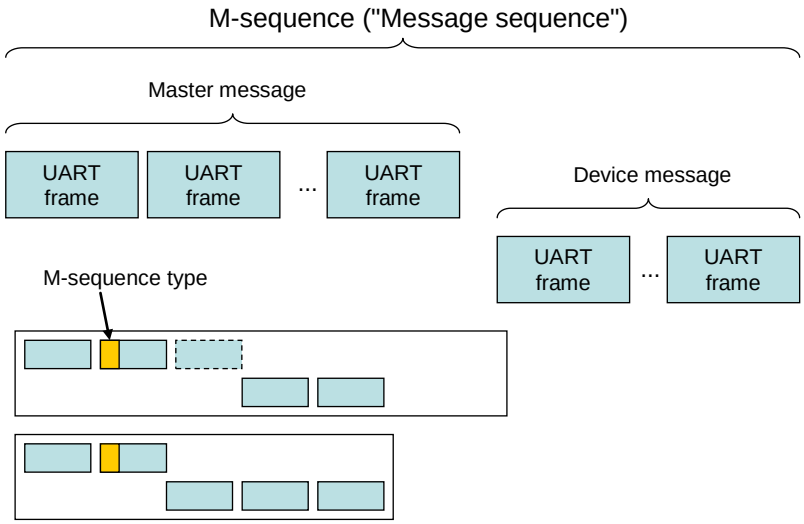
7.3.3.1 Généralités

Le rôle du module de traitement des messages est spécifié en 7.1 et 7.2.2.1. Le paragraphe 7.3.3 définit la structure et les types de séquences M ainsi que le comportement (dynamique) du module de traitement des messages.

7.3.3.2 Séquences M

Un Maître et son Dispositif échangent des données au moyen d'une séquence de messages (séquence M). Comme illustré en Figure 36, une séquence M comprend un message provenant du Maître, suivi d'un message provenant du Dispositif. Chaque message est constitué de trames UART.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
M-sequence ("Message sequence")	Séquence M ("séquence de Message")
UART frame	Trame UART
M-sequence type	Type de séquence M
Master message	Message du Maître
Device message	Message du Dispositif

Figure 36 – Séquences de messages SDCI

Tous les types de données à octets multiples doivent être transmis sous forme de séquence "gros-boutiste", c'est-à-dire que l'octet de poids fort (MSO) doit être envoyé en premier, suivi par des octets de poids faible en ordre décroissant, l'octet de poids faible (LSO) étant envoyé en dernier, comme illustré en Figure 2.

Le message du Maître commence par l'octet MC ("M-sequence Control" – "Contrôle de séquence M") suivi par l'octet CKT ("CHECK/TYPE" – Vérification/Type) et, de manière facultative, par des octets PD ("Process Data" – "Données de processus") et/ou des octets OD ("On-request Data" – "Données à la demande"). Le message de Dispositif commence optionnellement par des octets PD "Données de Processus" et/ou des octets OD "Données à la Demande", suivis par l'octet CKS ("CHECK/STAT").

Divers types de séquences M peuvent être sélectionnés pour répondre aux besoins particuliers d'un actionneur ou d'un capteur (vitesse de balayage, quantité de Données de Processus). La longueur des messages du Maître et du Dispositif peut varier en fonction du type de message et du sens de transmission des données, voir la Figure 36.

La Figure 37 donne un aperçu des types de séquences M définis. Les parties représentées par des lignes en pointillés dépendent du sens, lecture ou écriture, dans l'octet de contrôle de séquence M.

Les types de séquences M fixes sont le TYPE_0, le TYPE_1_1, le TYPE_1_2, ainsi que les TYPE_2_1 à TYPE_2_6. Les types de séquences M variables sont le TYPE_1_V et le TYPE_2_V.

Les différents types de séquences M répondent aux diverses exigences des capteurs et actionneurs en termes de taille de Données de Processus et de conditions correspondantes. Pour plus de détails sur les types de séquences M, voir A.2. Les contraintes temporelles applicables aux séquences M sont décrites au A.3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

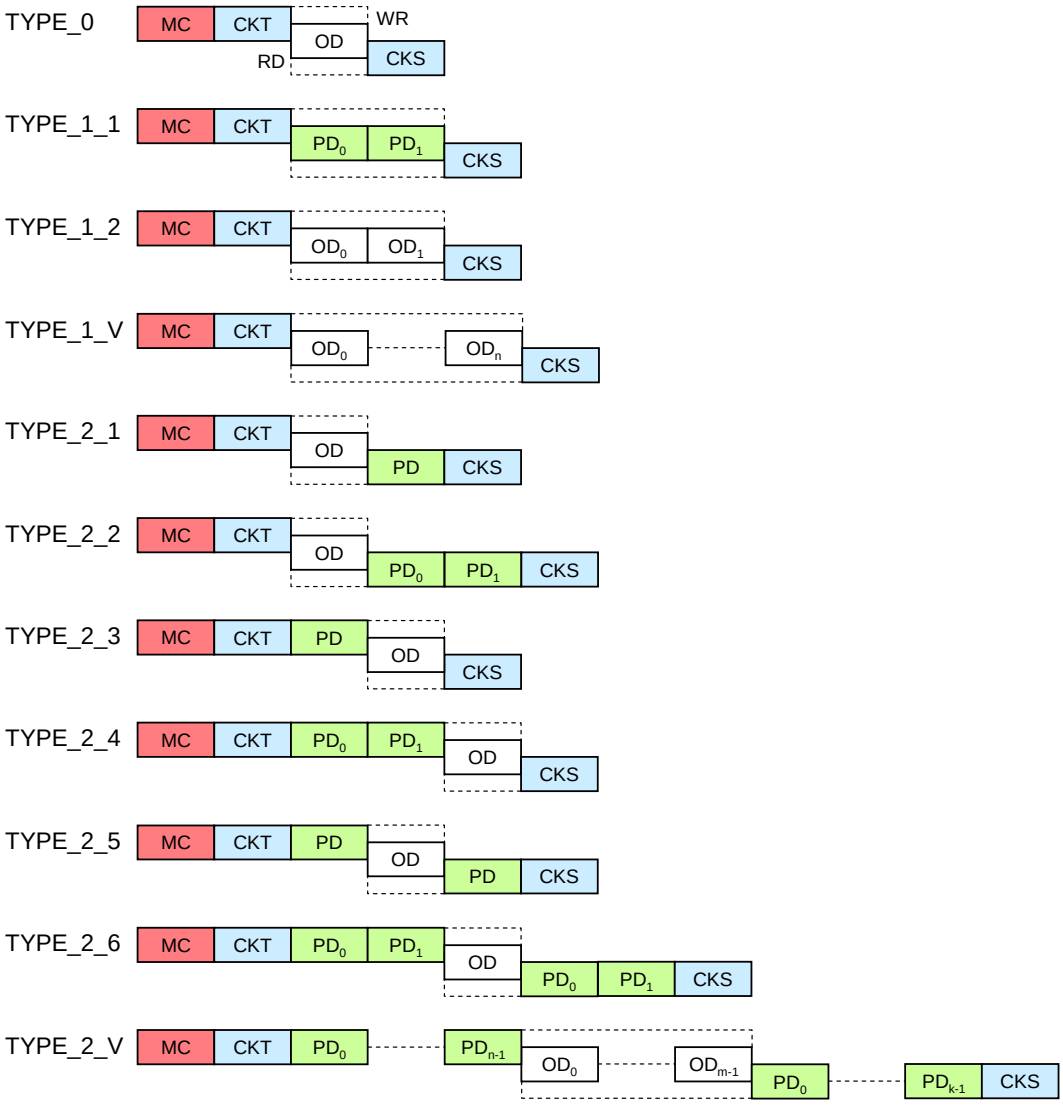


Figure 37 – Aperçu des types de séquence M

7.3.3.3 Contraintes liées au MasterCycleTime

À l'état STARTUP et PREOPERATE, un Dispositif est capable de communiquer de manière acyclique. Pour détecter la déconnexion des Dispositifs, il est fortement recommandé que le Maître effectue à partir de là une communication périodique ("message d'entretien") via des séquences M acycliques à travers la couche de liaison de données. Les durées minimales de récupération pour la communication acyclique, spécifiées en A.2.6, doivent être prises en compte.

Après ces phases, la communication cyclique de Données de Processus peut être lancée par le Maître via le service DL_SetMode (OPERATE). Les types de séquences M pour les échanges de données cycliques doivent être utilisés au cours de cette phase de communication pour échanger avec un Dispositif des Données de Processus (PD) et des Données à la Demande (voir le Tableau A.9 et le Tableau A.10).

Le Maître doit utiliser pour le temps t_{CYC} la valeur indiquée dans le paramètre de Dispositif "MasterCycleTime" (voir le Tableau B.1) avec une tolérance relative comprise entre 0 % et +10 % (y compris la gigue).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

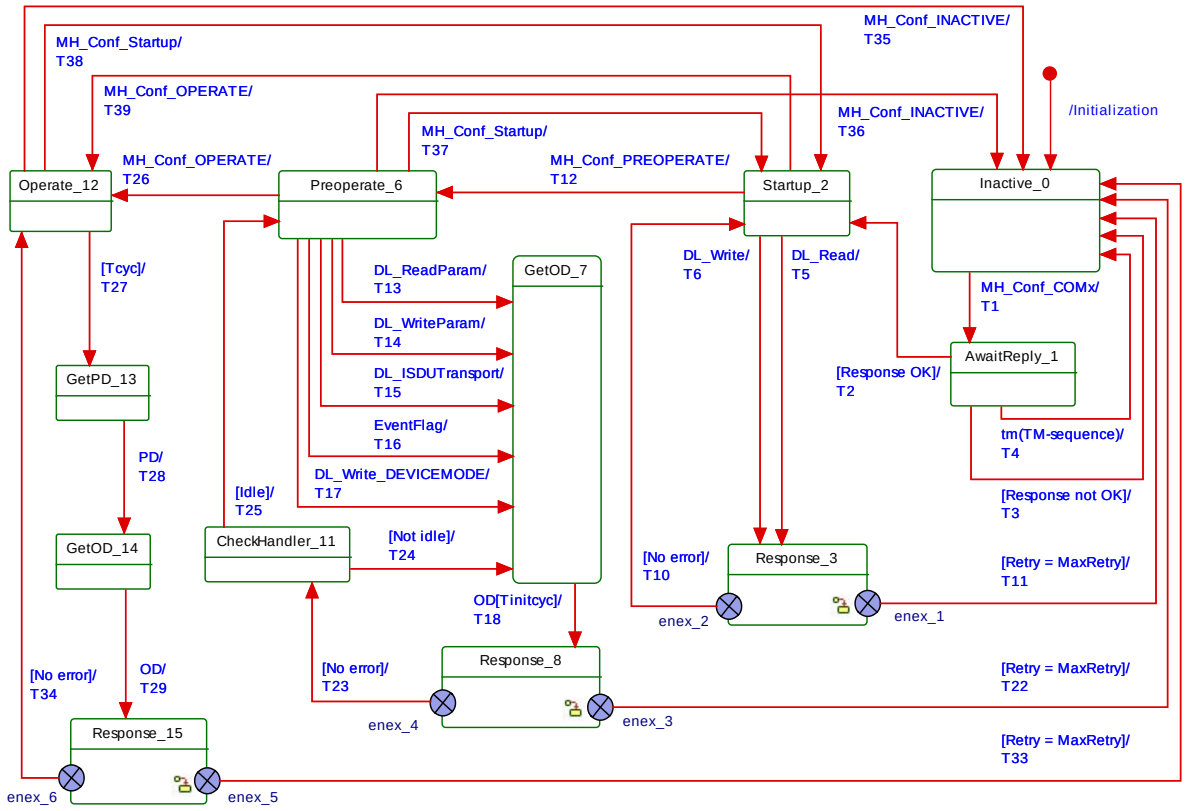
Dans les cas où le Dispositif est tenu d'être ramené au mode SIO après paramétrage, le Maître doit envoyer une commande de "Repli" (voir le Tableau B.2), suivie par une confirmation du Dispositif.

7.3.3.4 Diagramme d'états du module de traitement des messages du Maître

La Figure 38 illustre le diagramme d'états du module de traitement des messages du Maître. Trois sous-machines décrivant les réactions en cas d'erreurs de communication sont illustrées en Figure 39, en Figure 40 et en Figure 41.

Le module de traitement des messages prend en charge les exigences de communication particulières dans les états "EstablishCom", "Startup", "PreOperate" et "Operate" du module de traitement de mode DL.

Une invocation réglementaire interne MH_Conf_COMx à l'état "Inactive_0" force le module de traitement des messages à envoyer des messages "test" avec une séquence M de TYPE_0 et différentes vitesses de transmission (COM3, COM2 ou COM1) au cours de la séquence d'établissement de communication.



Initialization	Initialisation
Response OK	Réponse positive
Response not ok	Réponse négative
Idle	Repos
Retry	Nouvelle tentative

Figure 38 – Diagramme d'états du module de traitement des messages du Maître

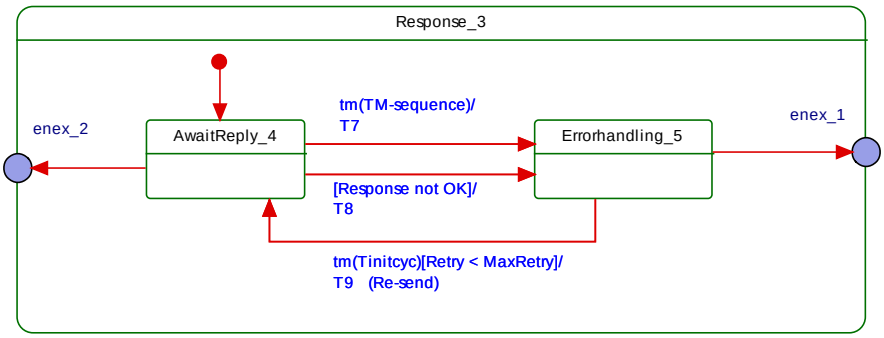
L'état "Startup_2" fournit tous les moyens de communication nécessaires à la prise en charge des contrôles d'identité de la gestion système à l'aide des services DL_Read et DL_Write. Le module de traitement des messages attend l'invocation de ces services pour envoyer et recevoir des messages (communication acyclique).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

L'état "Preoperate_6" est le point de reprise de toutes les activités relatives aux Données à la Demande, telles que les ISDU, les commandes et les Événements pour le paramétrage du Dispositif. Le module de traitement des messages attend l'invocation des services illustrés en Figure 38 pour envoyer et recevoir des messages (communication acyclique).

L'état "Operate_12" est le point de reprise de l'échange cyclique de Données de Processus. En fonction du type de séquence M, le module de traitement des messages génère des messages du Maître avec les Données de Processus acquises auprès du module de traitement des Données de Processus, via le service PD, et optionnellement les Données à la Demande acquises auprès du module de traitement des Données à la Demande, via le service OD.

La Figure 39 illustre la sous-machine de l'état "Response 3".

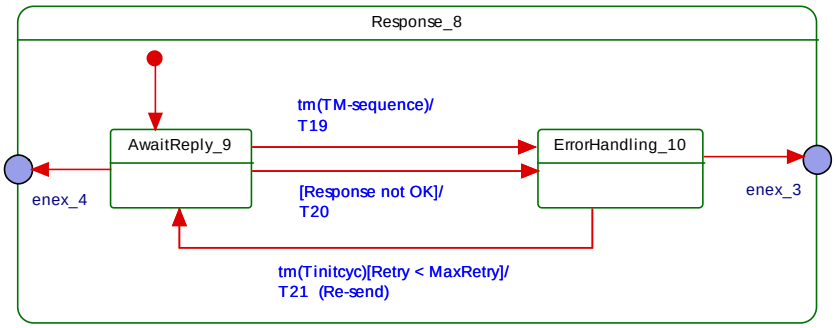


Légende

Anglais	Français
Response not ok	Réponse négative

Figure 39 – Sous-machine "Response 3" du module de traitement des messages

La Figure 40 illustre la sous-machine de l'état "Response 8".



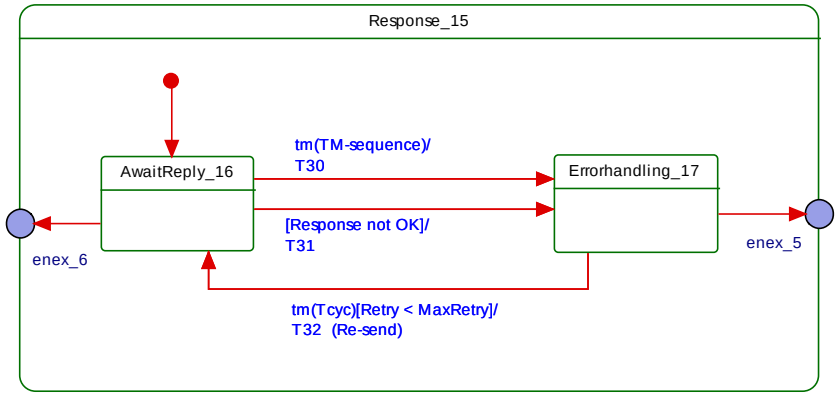
Légende

Anglais	Français
Response not ok	Réponse négative
Re-send	Envoi de nouveau

Figure 40 – Sous-machine "Response 8" du module de traitement des messages

La Figure 41 illustre la sous-machine de l'état "Response 15".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Response not ok	Réponse négative
Retry	Nouvelle tentative
Re-send	Envoi de nouveau

Figure 41 – Sous-machine "Response 15" du module de traitement des messages

Le Tableau 44 présente les tables de transitions d'états du module de traitement des messages du Maître.

Tableau 44 – Table de transitions d'états du module de traitement des messages du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente de demande d'un message "test" via l'invocation de MH_Conf_COMx (voir la Figure 34 et le Tableau 42) en provenance du module de traitement de mode DL.
AwaitReply_1	Attente de réponse du Dispositif au message "test". Retour à l'état Inactive_0 si le temps de séquence TM s'est écoulé sans réponse du Dispositif ou si la réponse au message "test" n'a pas pu être décodée. En cas de réponse correcte du Dispositif, le module de traitement des messages passe à l'état Startup_2.
Startup_2	Lorsque cet état est atteint via la transition T2, il est chargé de contrôler l'échange acyclique de Données à la Demande selon les conditions spécifiées dans le Tableau A.7. Tout service DL_Write ou DL_Read en provenance de la Gestion Système génère une transition.
Response_3	Le service OD a entraîné l'envoi d'un message correspondant par le module de traitement de message. Dans ce pseudo-état, la sous-machine attend la réponse et vérifie qu'elle est correcte.
SM: AwaitReply_4	Cet état vérifie que le temps de séquence T_M a expiré et que la réponse est correcte.
SM: ErrorHandling_5	Si elle est incorrecte, le module de traitement des messages renverra le message au bout d'un temps d'attente $T_{initcyc}$. Si le nombre de nouvelles tentatives est trop élevé, le module de traitement des messages passera à l'état Inactive_0 .
Preoperate_6	À la réception d'une invocation MH_Conf_PREOPERATE, le module de traitement des messages est passé dans cet état. Le module de traitement des messages est à présent chargé de contrôler l'échange acyclique de Données à la Demande selon les conditions spécifiées dans le Tableau A.8. Tout service DL_ReadParam, DL_WriteParam, DL_ISDUTransport, DL_Write ou EventFlag génère une transition.
GetOD_7	Le module de traitement des messages a utilisé le service ODTrig pour acquérir des OD auprès du module de traitement de Données à la Demande. Le module de traitement des messages attend le service OD pour envoyer un message au bout d'une période $T_{initcyc}$.
Response_8	Le service OD a entraîné l'envoi d'un message correspondant par le module de traitement des messages. Dans ce pseudo-état, la sous-machine attend la réponse et vérifie qu'elle est correcte.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
SM: AwaitReply_9	Cet état permet de vérifier si la séquence temporelle T_M s'est écoulée et si la réponse est correcte.
SM: ErrorHandling_10	Si elle est incorrecte, le module de traitement des messages renverra le message au bout d'un temps d'attente $T_{initcyc}$. Si le nombre de nouvelles tentatives est trop élevé, le module de traitement des messages passera à l'état Inactive_0.
CheckHandler_11	Certains services nécessitent plusieurs cycles d'acquisition d'OD pour échanger les OD. Lorsque le module de traitement des OD, des ISDU ou d'Événements concerné est revenu à l'état au repos (inactif), le module de traitement des messages peut quitter la boucle d'acquisition des OD.
Operate_12	À la réception d'une invocation MH_Conf_OPERATE, le module de traitement des messages est passé dans cet état, et au bout d'un temps initial $T_{initcyc}$, ce dernier est chargé de contrôler l'échange cyclique de Données de Processus et de Données à la Demande selon les conditions spécifiées dans le Tableau A.9 et le Tableau A.10. Le module de traitement des messages relance de lui-même un nouveau cycle de messages cycle après expiration de la période t_{CYC}.
GetPD_13	Le module de traitement des messages a utilisé le service PDTrig pour acquérir des PD auprès du module de traitement des Données de Processus. Le module de traitement des messages attend le service PD puis passe à l'état GetOD_14.
GetOD_14	Le module de traitement des messages a utilisé le service ODTrig pour acquérir des OD auprès du module de traitement de Données à la Demande. Le module de traitement des messages attend le service OD pour compléter les PD déjà acquises et envoyer un message contenant les PD/OD acquises.
Response_15	Le module de traitement des messages a envoyé un message contenant les PD/OD acquises. Dans ce pseudo-état, la sous-machine attend la réponse et vérifie qu'elle est correcte.
SM: AwaitReply_16	Cet état permet de vérifier si la séquence temporelle T_M s'est écoulée et si la réponse est correcte.
SM: ErrorHandling_17	Si elle est incorrecte, le module de traitement des messages renverra le message au bout d'un temps d'attente t_{CYC}. Si le nombre de nouvelles tentatives est trop élevé, le module de traitement des messages passera à l'état Inactive_0.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	Envoi d'un message à la vitesse de transmission COMx demandée et avec une séquence M de TYPE_0: Lecture de la page 1 de Paramètres Directs, adresse 0x02 ("MinCycleTime"), compilation en un contrôle de séquence M MC = 0xA2 (voir A.1.2). Démarrage du temporisateur avec la séquence T_M.
T2	1	2	Renvoi de la valeur de "MinCycleTime" via la confirmation de service DL_Read.
T3	1	0	Réinitialisation du temporisateur (séquence T_M).
T4	1	0	Réinitialisation du temporisateur (séquence T_M).
T5	2	3	Envoi d'un message en utilisant la vitesse de transmission établie, le canal de communication de page, et l'option d'accès en lecture (voir A.1.2). Démarrage du temporisateur avec la séquence T_M.
T6	2	3	Envoi d'un message en utilisant la vitesse de transmission établie, le canal de communication de page, et l'option d'accès en écriture (voir A.1.2). Démarrage du temporisateur avec la séquence T_M.
T7	4	5	Réinitialisation du temporisateur (séquence T_M).
T8	4	5	Réinitialisation du temporisateur (séquence T_M).
T9	5	4	Retransmission du message au bout d'un temps $T_{initcyc.}$ Redémarrage du temporisateur avec la séquence T_M.
T10	3	2	Renvoi de la confirmation de service DL_Read ou DL_Write respectivement à la Gestion Système.
T11	3	0	Le module de traitement des messages renvoie MH_Info (COMLOST) au module de traitement de mode DL.
T12	2	6	-

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

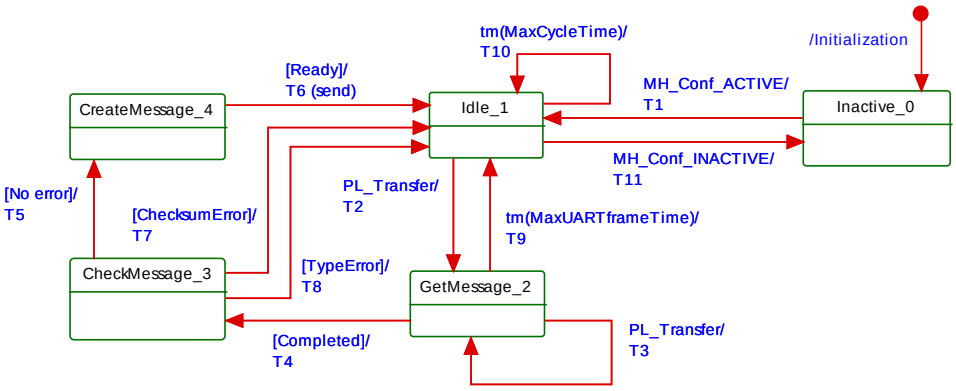
DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
T13	6	7	Le module de traitement des messages invoque le service ODTrig pour le module de traitement des Données à la Demande (voir la Figure 46), qui est à l'état "ISDU_1". Dans cet état, il force le module de traitement des ISDU à fournir le service OD en correspondance au service DL_ReadParam (voir la Figure 49, Transition T13).
T14	6	7	Le module de traitement des messages invoque le service ODTrig pour le module de traitement des Données à la Demande (voir la Figure 46), qui est à l'état "ISDU_1". Dans cet état, il force le module de traitement des ISDU à fournir le service OD en correspondance au service DL_WriteParam (voir la Figure 49, Transition T13).
T15	6	7	Le module de traitement des messages invoque le service ODTrig pour le module de traitement des Données à la Demande (voir la Figure 46), qui est à l'état "ISDU_1". Dans cet état, il force le module de traitement des ISDU à fournir le service OD en correspondance au service DL_ISDUTransport (voir la Figure 49, Transition T2). Le module de traitement des messages peut nécessiter plusieurs cycles avant que le module de traitement des ISDU ne retourne à l'état de "repos".
T16	6	7	Le module de traitement des messages invoque le service ODTrig pour le module de traitement des Données à la Demande (voir la Figure 46), qui est à l'état "Event_4". Dans cet état, il force le module de traitement d'Événements à fournir le service OD en correspondance au service EventFlag (voir la Figure 53, Transition T2). Le module de traitement des messages peut nécessiter plusieurs cycles avant que le module de traitement d'Événements ne retourne à l'état de "repos".
T17	6	7	Le module de traitement des messages invoque le service ODTrig pour le module de traitement des Données à la Demande (voir la Figure 46), qui est à l'état "ISDU_1". Dans cet état, il force le module de traitement des ISDU à fournir le service OD en correspondance au service DL_Write (voir la Figure 49, Transition T13).
T18	7	8	Envoi d'un message au bout d'un temps de récupération $T_{initcyc}$ généré par le service OD.req. Démarrage du temporisateur avec la séquence T_M.
T19	9	10	Réinitialisation du temporisateur (séquence T_M).
T20	9	10	Réinitialisation du temporisateur (séquence T_M).
T21	10	9	Retransmission du message au bout d'un temps $T_{initcyc}$. Redémarrage du temporisateur avec la séquence T_M.
T22	8	0	Le module de traitement des messages passe à l'état Inactive_0 et renvoie MH_Info (COMLOST) au module de traitement de DL-mode.
T23	8	11	-
T24	11	7	Acquisition des OD par invocation du service ODTrig vers le module de traitement de Données à la Demande, qui à son tour déclenche le module de traitement qui en a actuellement la charge, via l'invocation de ISDU ou d'EventTrig.
T25	11	6	Renvoi du résultat via la primitive de service OD.cnf
T26	6	12	Le module de traitement des messages passe à l'état Operate_12.
T27	12	13	Démarrage du temporisateur t_{CYC}. Acquisition des PD par invocation du service PDTrig vers le module de traitement des Données de Processus (voir la Figure 44).
T28	13	14	Acquisition des OD par invocation du service ODTrig vers le module de traitement des Données à la Demande (voir la Figure 46).
T29	14	15	PD et OD prêtes par l'intermédiaire du service PD.req du module de traitement des PD et du service OD.req via le module de traitement des OD. Le module de traitement des messages envoie le message. Démarrage du temporisateur avec la séquence T_M.
T30	16	17	Réinitialisation du temporisateur (séquence T_M).
T31	16	17	Réinitialisation du temporisateur (séquence T_M).
T32	17	16	Retransmission du message au bout d'un temps t_{CYC}. Redémarrage du temporisateur avec la séquence T_M.
T33	15	0	Le module de traitement des messages passe à l'état Inactive_0 et renvoie MH_Info (COMLOST) au module de traitement de DL-mode.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
T34	15	12	Le message de réponse du Dispositif est correct. Renvoi de PD via le service PD.cnf et via l'invocation de PDTrig vers le module de traitement des PD (voir le Tableau 46). Renvoi des OD via le service OD.cnf et via l'invocation de ODTrig vers le module de traitement de Données à la Demande, qui les redirige vers le module de traitement des ISDU (voir le Tableau 51), le module de traitement des commandes (voir le Tableau 54), ou le module de traitement d'Événements (voir le Tableau 57) concerné.
T35	12	0	Le module de traitement des messages passe à l'état Inactive_0 et renvoie MH_Info (COMLOST) au module de traitement de DL-mode.
T36	6	0	Le module de traitement des messages passe à l'état Inactive_0 et renvoie MH_Info (COMLOST) au module de traitement de DL-mode.
T37	6	2	-
T38	12	2	-
T39	2	12	-
ÉLÉMENTS INTERNES		**TYPE**	**DÉFINITION**
Retry (Nouvelle tentative)		Variable	Compteur du nombre de tentatives
MaxRetry		Constant	MaxRetry = 2, voir le Tableau 97
$t_{M\text{-sequence}}$		Temps	Voir l'Equation (A.6)
t_{CYC}		Temps	Le service DL_SetMode fournit cette valeur avec son paramètre "M-sequenceTime". Voir l'Equation (A.7)
$t_{initcyc}$		Temps	Voir A.2.6
MH_Conf_xxx		Invocation	Voir le Tableau 42

7.3.3.5 Diagramme d'états du module de traitement des messages du Dispositif

La Figure 42 illustre le diagramme d'états du module de traitement des messages du Dispositif.



Légende

Anglais	Français
Initialization	Initialisation
Ready	Prêt
Send	Emission
No error	Pas d'erreur
Completed	"Achevé"

Figure 42 – Diagramme d'états du module de traitement des messages du Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau 45 présente les tables de transitions d'états du module de traitement des messages du Dispositif.

Tableau 45 – Tables de transitions d'états du module de traitement des messages du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente de l'activation par le module de traitement de mode DL du Dispositif par l'intermédiaire de MH_Conf_ACTIVE (voir le Tableau 43, Transition T1).
Idle_1	Attente de la première trame UART du message de Maître par l'intermédiaire de l'indication de service PL_Transfer. Vérification de l'expiration du temps "MaxCycleTime".
GetMessage_2	Réception d'une trame UART de message de Maître. Vérification du nombre de trames UART reçues (Le Dispositif connaît le type de séquence M et il connaît donc le nombre de trames UART). Vérification de l'expiration du temps "MaxUARTframeTime" .
CheckMessage_3	Vérification du type de séquence M et de la somme de contrôle du message reçu.
CreateMessage_4	Compilation du message à partir des services OD.rsp, PD.rsp, EventFlag et PDStatus.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	2	Démarrage de "MaxUARTframeTime" et "MaxCycleTime" en l'état OPERATE.
T3	2	2	Redémarrage du temporisateur "MaxUARTframeTime".
T4	2	3	Redémarrage du temporisateur "MaxUARTframeTime".
T5	3	4	Invocation des indications de service OD.ind et PD.ind
T6	4	1	Compilation et invocation de la réponse de service PL_Transfer.rsp (Le Dispositif envoie le message de réponse)
T7	3	1	Indication d'erreur au module de traitement de mode DL via MHInfo (CHECKSUM_MISMATCH)
T8	3	1	Indication d'erreur au module de traitement de mode DL via MHInfo (ILLEGAL_MESSAGETYPE)
T9	2	1	Réinitialisation des deux temporisateurs "MaxUARTframeTime" et "MaxCycleTime".
T10	1	1	Indication d'erreur au module de traitement de mode DL via MHInfo (COMLOST). Les actionneurs doivent observer cette information et prendre les mesures correspondantes (voir 10.2 et 10.7.3).
T11	1	0	Le module de traitement des messages du Dispositif passe à l'état Inactive_0.

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
MaxUARTFrameTime	Temps	Temps de transmission d'une trame UART (11 T_{BIT}) plus la valeur maximale de t_1 (1 T_{BIT}) = 11 T_{BIT}.
MaxCycleTime	Temps	Le but du temporisateur "MaxCycleTime" est de vérifier si l'échange cyclique de Données de Processus a pris trop de temps ou s'il a été interrompu. Le MaxCycleTime doit être > au MasterCycleTime (voir A.3.7).
TypeError	Protection	Une des détections d'erreurs possibles: ILLEGAL_MESSAGETYPE, ou COMLOST
ChecksumError	Protection	Erreur de somme de contrôle de message détectée

7.3.4 Module de traitement des Données de Processus

7.3.4.1 Généralités

Le transport des Données de Processus de sortie est réalisé en utilisant les services DL_OutputUpdate et celui des Données de Processus d'entrée en utilisant les services DL_InputTransport (voir la Figure 26). Un cycle de Données de Processus est terminé lorsque

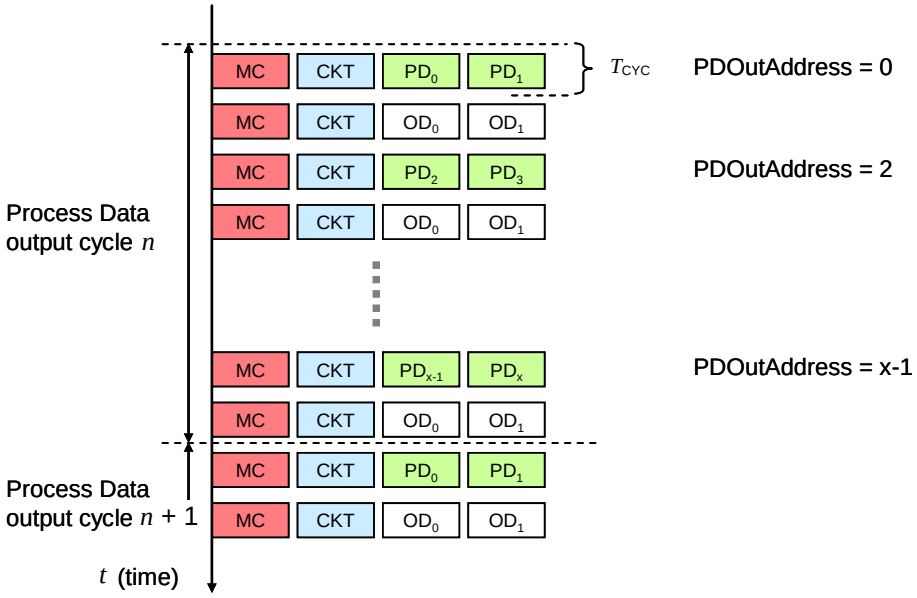
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

l'ensemble des Données de Processus a été transféré entre le Maître et le Dispositif dans le sens demandé. Ce cycle peut durer pendant plusieurs séquences M.

Toutes les Données de Processus sont transmises dans une séquence M lorsque des séquences M de TYPE_2_x (voir la Figure 37) sont utilisées. Dans ce cas, la durée d'exécution d'un cycle de Données de Processus est égale à la durée de cycle t_{CYC}.

7.3.4.2 Mode d'entrelacement

Toutes les Données de Processus et les Données à la Demande sont transmises dans ce cas en utilisant plusieurs séquences M alternantes de TYPE_1_1 (Données de Processus) et de TYPE_1_2 (Données à la Demande), comme illustré en Figure 43. Cette figure montre l'écriture de messages du Maître contenant des Données de Processus de sortie vers un Dispositif. Le paramètre de service PDOutAddress indique le partitionnement des PD de sortie à transmettre (voir 7.2.2.3). Pour les Données de Processus d'entrée, le paramètre de service PDInAddress indique quant à lui le partitionnement des PD d'entrée. Dans un cycle de Données de Processus, toutes les PD d'entrée doivent être lues en premier, puis suivies par toutes les PD de sortie à écrire. Un cycle de Données de Processus comprend toutes les durées de cycle nécessaires à la transmission de l'ensemble des Données de Processus.



Légende

Anglais	Français
Cycle time	Durée de cycle
Process Data output cycle *n*	Cycle *n* de Données de Processus de sortie
Process Data output cycle *n*+1	Cycle *n*+1 de Données de Processus de sortie
Not recommended for new developments of Devices	Non recommandé pour des Dispositifs nouvellement développés
t (time)	*t* (temps)

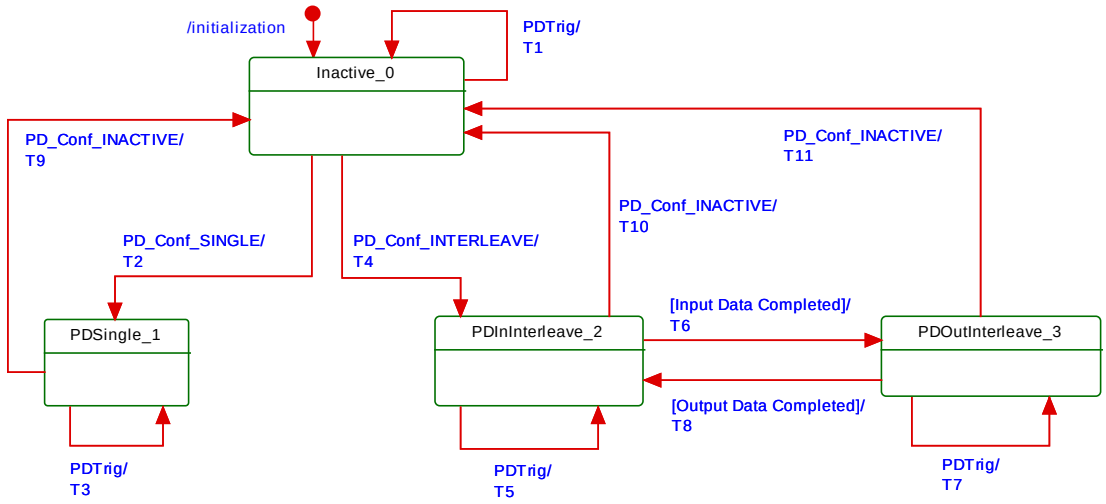
Figure 43 – Mode d'entrelacement pour la transmission segmentée des Données de Processus

Le mode d'entrelacement est destiné aux seuls Dispositifs patrimoniaux.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.3.4.3 Diagramme d'états du module de traitement des Données de Processus du Maître

La Figure 44 illustre le diagramme d'états du module de traitement des Données de Processus du Maître.



Légende

Anglais	Français
Initialization	Initialisation
Input Data Completed	Données d'entrée achevées
Output Data Completed	Données de sortie achevées

Figure 44 – Diagramme d'états du module de traitement des Données de Processus du Maître

Le Tableau 46 présente les tables de transitions d'états du module de traitement des Données de Processus du Maître.

Tableau 46 – Tables de transitions d'états du module de traitement des Données de Processus du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
PDSingle_1	Communication des Données de Processus en une séquence M unique
PDInInterleave_2	Communication des Données de Processus d'entrée en mode d'entrelacement
PDOutInterleave_3	Communication des Données de Processus de sortie en mode d'entrelacement

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	0	Invocation de PD.req sans Données de Processus
T2	0	1	NOTE Le module de traitement de mode DL a configuré le module de traitement des Données de Processus pour une transmission de PD en une seule fois (voir le Tableau 42, T10 ou T11)
T3	1	1	Récupération de données auprès du service DL_PDOutputUpdate et invocation de PD.req pour propager les PD résultantes au module de traitement des messages. Récupération de données à partir de PD.cnf et invocation de DL_PDInputTransport.ind et DL_PDCycle.ind pour propager les PD

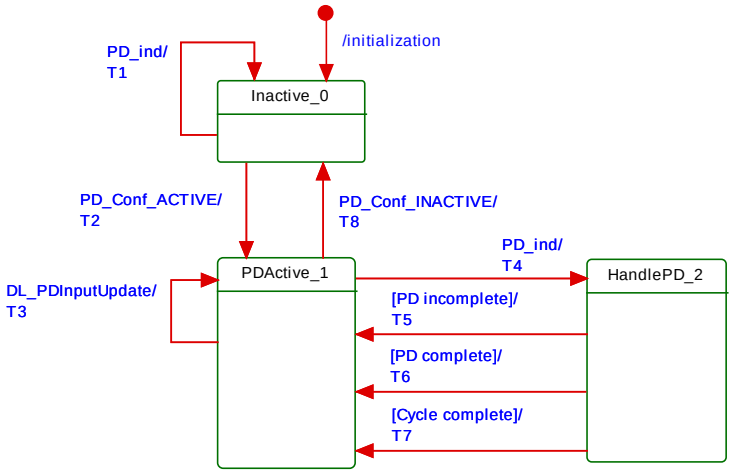
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
			d'entrée vers la couche AL.
T4	0	2	NOTE Configuré pour transmission des PD en entrelacement (voir le Tableau 42, T10 ou T11).
T5	2	2	Invocation de PD.req et utilisation de PD.cnf pour préparer DL_PDInputTransport.ind.
T6	2	3	Invocation de DL_PDInputTransport.ind et de DL_PDCycle.ind pour propager les PD d'entrée vers la couche AL (voir 7.2.1.11).
T7	3	3	Récupération de données auprès du service DL_PDOutputUpdate et invocation de PD.req pour propager les PD de sortie au module de traitement des messages.
T8	3	2	Invocation de DL_PDCycle.ind pour indiquer la fin du cycle de Données de Processus à la couche AL (voir 7.2.1.12).
T9	1	0	-
T10	2	0	-
T11	3	0	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
<Aucun>		

7.3.4.4 Diagramme d'états du module de traitement des Données de Processus du Dispositif

La Figure 45 illustre le diagramme d'états du module de traitement des Données de Processus du Dispositif.



Légende

Anglais	Français
Initialization	Initialisation
PD incomplete	PD incomplètes
PD complete	PD complètes
Cycle complete	Cycle complet

Figure 45 – Diagramme d'états du module de traitement des Données de Processus du Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Dans ce contexte, se reporter aux diagrammes de séquences en Figure 65 et Figure 66.

Le Tableau 47 présente les tables de transitions d'états du module de traitement des Données de Processus du Dispositif.

Tableau 47 – Tables de transitions d'états du module de traitement des Données de Processus du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
PDActive_1	Module de traitement actif et en attente de la demande suivante du module de traitement des messages via le service PD ou le service DL_PDInputUpdate en provenance de la couche AL.
HandlePD_2	Vérification de l'exhaustivité des Données de Processus en mode d'entrelacement

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	0	Les Données de Processus sont ignorées
T2	0	1	-
T3	1	1	Préparation des Données de Processus d'entrée pour PD.rsp à la demande suivante du module de traitement des messages
T4	1	2	Le module de traitement des messages demande des PD d'entrée via un service PD.ind et remet des PD de sortie ou des segments de PD de sortie. Invocation de PD.rsp avec des Données de Processus d'entrée en mode sans entrelacement (voir 7.2.2.3).
T5	2	1	-
T6	2	1	Invocation de DL_PDOutputTransport.ind (voir 7.2.1.9)
T7	2	1	Invocation de DL_PDCycle.ind (voir 7.2.1.12)
T8	1	0	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
PD_ind	Étiquette	Le service PD.ind a été invoqué par le module de traitement des messages

7.3.5 Module de traitement de Données à la Demande

7.3.5.1 Généralités

Le module de traitement de Données à la Demande du Maître est un diagramme d'états subordonné actif à l'état "Startup_2", "PreOperate_3" et "Operate_4" du module de traitement de mode DL (voir la Figure 33). Il contrôle trois autres diagrammes d'états appelés module de traitement des ISDU, module de traitement des commandes et module de traitement d'Événements. Par défaut, il commence toujours en module de traitement des ISDU.

Dès qu'un EventFlag.ind est reçu, le diagramme d'états passe en module de traitement d'Événements. Après avoir lu entièrement les informations d'Événement, il retourne à l'état de module de traitement des ISDU.

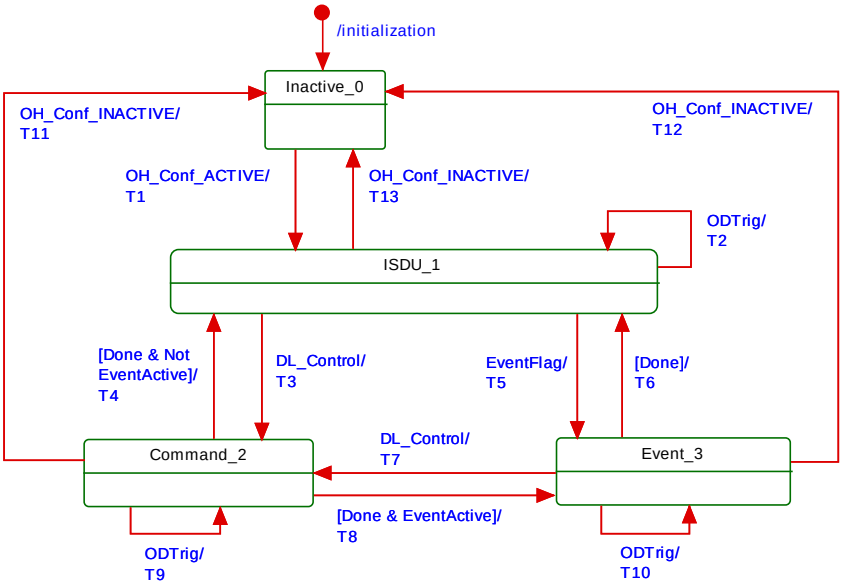
Dès qu'un service DL_Control.req ou PDInStatus.ind est reçu lorsqu'il est en module de traitement des ISDU ou en module de traitement d'Événements, le diagramme d'états passe en module de traitement des commandes. Une fois la commande servie, le diagramme d'états retourne à l'état précédemment actif (ISDU ou Événement).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.3.5.2 Diagramme d'états du module de traitement de Données à la Demande du Maître

La Figure 46 illustre le diagramme d'états du module de traitement de Données à la Demande du Maître.

Le module de traitement de Données à la Demande redirige la primitive de service ODTrig.ind pour le contenu du message suivant vers le module de traitement subsidiaire couramment actif (ISDU, commandes ou Événements). Ceci est réalisé par l'intermédiaire d'une des invocations ISDUTrig, CommandTrig ou EventTrig.



Légende

Anglais	Français
Initialization	Initialisation
Done & Not Event Active	Effectué & Pas de EventActive
Done	Effectué
Done & Event Active	Effectué & EventActive

Figure 46 – Diagramme d'états du module de traitement de Données à la Demande du Maître

Le Tableau 48 présente les tables de transitions d'états du module de traitement des Données à la Demande du Maître.

Tableau 48 – Tables de transitions d'états du module de traitement des Données à la Demande du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
ISDU_1	État par défaut du module de traitement de Données à la Demande (non prioritaire)
Command_2	État pour le contrôle du Dispositif via des commandes prioritaires
Event_3	État pour l'acheminement des informations d'Événements (erreurs, avertissements, notifications) prioritaires

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	1	Le module de traitement de Données à la Demande propage le service ODTrig.ind à présent désigné ISDUTrig vers le module de traitement des ISDU (voir la Figure 49). Dans le cas des services DL_Read, DL_Write, DL_ReadParam ou DL_WriteParam, le module de traitement des ISDU utilise une transition séparée (voir la Figure 49, T13).
T3	1	2	-
T4	2	1	-
T5	1	3	EventActive = VRAI
T6	3	1	EventActive = FAUX
T7	3	2	-
T8	2	3	-
T9	2	2	Le module de traitement de Données à la Demande propage le service ODTrig.ind à présent désigné CommandTrig vers le module de traitement des commandes (voir la Figure 51)
T10	3	3	Le module de traitement de Données à la Demande propage le service ODTrig.ind à présent désigné EventTrig vers le module de traitement d'Événements (voir la Figure 53)
T11	2	0	-
T12	3	0	-
T13	1	0	-
ÉLÉMENTS INTERNES		**TYPE**	**DÉFINITION**
EventActive		Bool	Drapeau indiquant le sens de reprise après une interruption du traitement d'Événements pour une demande de commande prioritaire

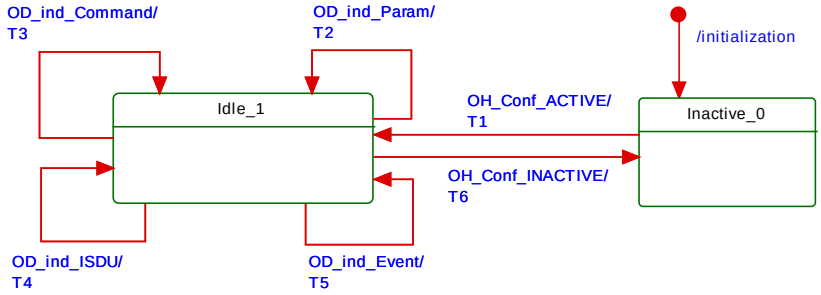
7.3.5.3 Diagramme d'états du module de traitement de Données à la Demande du Dispositif

La Figure 47 illustre le diagramme d'états du module de traitement de Données à la Demande du Dispositif.

Le module de traitement de Données à la Demande du Dispositif obtient des informations sur le canal de communication et le paramètre ou l'adresse de FlowCTRL via le service OD.ind. Les canaux de communication sont totalement indépendants. Si l'accès est valide, le diagramme d'états correspondant (des ISDU, des commandes ou des Événements) est sollicité via le canal de communication associé.

Le Dispositif doit répondre aux demandes de lecture vers des plages d'adresses non utilisées de valeur "0". Il doit ignorer les demandes d'écriture vers des plages d'adresses non utilisées.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
initialization	Initialisation

Figure 47 – Diagramme d'états du module de traitement de Données à la Demande du Dispositif

Dans le cas d'un accès ISDU dans un Dispositif sans prise en charge des ISDU, le Dispositif doit répondre par "Pas de Service" (voir le Tableau A.12). Il n'est pas généré de message d'erreur.

NOTE OD.ind (R, ISDU, FlowCTRL = IDLE) est le message par défaut s'il n'y a pas de Données à la Demande en attente de transmission.

Le Tableau 49 présente les tables de transitions d'états du module de traitement des Données à la Demande du Dispositif.

Tableau 49 – Tables de transitions d'états du module de traitement des Données à la Demande du Dispositif

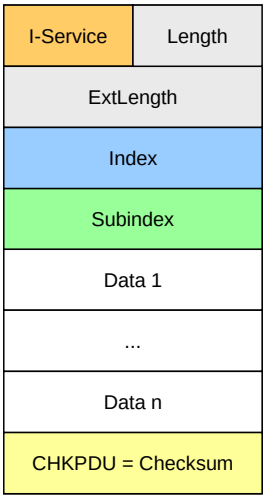
DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
Inactive_0		Attente d'activation	
Idle_1		Attente de messages avec des Données à la Demande via l'indication de service OD. Décomposition et analyse.	
TRANSITION	**ÉTAT DE LA SOURCE**	**ÉTAT DE LA CIBLE**	**ACTION**
T1	0	1	-
T2	1	1	Redirection vers le module de traitement des ISDU (page de Paramètres Directs).
T3	1	1	Redirection vers le module de traitement des commandes
T4	1	1	Redirection vers le module de traitement des ISDU
T5	1	1	Redirection vers le module de traitement d'Événements
T6	1	0	-
ÉLÉMENTS INTERNES		**TYPE**	**DÉFINITION**
OD_ind_Param		Service	Alias du service OD.ind (Lecture/Écriture, PAGE, 16 à 31, Données) dans le cas de DL_ReadParam ou DL_WriteParam
OD_ind_Command		Service	Alias du service OD.ind (Ecriture, PAGE, 0, MasterCommand)
OD_ind_ISDU		Service	Alias du service OD.ind (Lecture/Écriture, ISDU, FlowCtrl, Données)
OD_ind_Event		Service	Alias du service OD.ind (Lecture/Écriture, DIAGNOSTIC, n, Données)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.3.6 Module de traitement des ISDU

7.3.6.1 Unité de données de service indexée (ISDU)

La structure générale d'une ISDU est illustrée dans la Figure 48 et décrite de manière détaillée en A.5.



Légende

Anglais	Français
I-Service	"Service I"
Length	Longueur
ExtLength	ExtLength (longueur étendue)
Index	Index
Subindex	Sous-index
Data	Donnée
CHKPDU = Checksum	CHKPDU = Somme de contrôle

Figure 48 – Structure de l'ISDU

La séquence des éléments correspond à la séquence de transmission. Les éléments d'une ISDU peuvent prendre des formes diverses en fonction du type de "Service I" (voir A.5.2 et le Tableau A.12).

L'ISDU permet d'accéder à des objets de données (paramètres et commandes) à transmettre (voir la Figure 5) Les objets de données doivent être adressés par l'élément “Index”.

Tous les types de données à octets multiples doivent être transmis sous forme de séquence "gros-boutiste", c'est-à-dire que l'octet de poids fort (MSO) doit être envoyé en premier, suivi par des octets de poids faible en ordre décroissant, l'octet de poids faible (LSO) étant envoyé en dernier, comme illustré en Figure 2.

7.3.6.2 Transmission des ISDU

Une ISDU est transmise via le canal de communication des ISDU (voir la Figure 7 et A.1.2). Un certain nombre de messages est en général requis pour effectuer cette transmission (segmentation). Le Maître transfère une ISDU en envoyant une demande de "Service I" (Lecture/Écriture) au Dispositif via le canal de communication des ISDU. Il reçoit ensuite la réponse du Dispositif par le même canal.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Sur le canal de communication des ISDU, l'élément "Adresse" dans l'octet de contrôle de séquence M inclut un compteur (= FlowCTRL). FlowCTRL contrôle le flux de données segmentées (voir A.1.2) en comptant les éléments de l'ISDU (modulo 16) pendant la transmission.

Le Maître utilise l'élément "Longueur" de l'ISDU et FlowCTRL pour vérifier l'accomplissement de la transmission complète.

Les valeurs admissibles de FlowCTRL sont spécifiées dans le Tableau 50.

Tableau 50 – Définitions du FlowCTRL

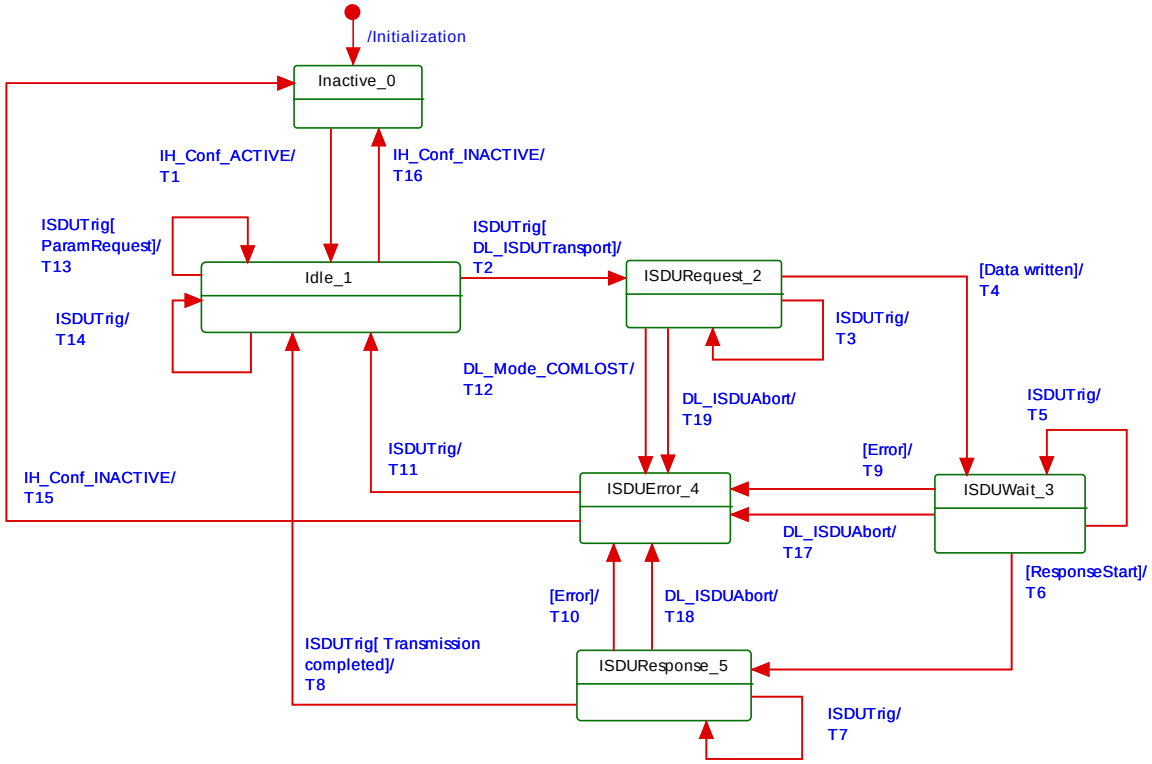
FlowCTRL	Définition
0x00 à 0x0F	COUNT Compteur d'éléments dans une ISDU. S'incrémente à partir de 1 après un DÉBUT D'ISDU. Revient de 15 à 0 en cas d'Événement de débordement.
0x10	START Début d'un "Service I" de ISDU, c'est-à-dire début d'une demande ou d'une réponse. Pour le début d'une demande, tous les services précédemment incomplets peuvent être rejetés. Pour une demande de démarrage associée à une réponse, un Dispositif doit envoyer “Pas de Service” jusqu'à ce que son application renvoie des données de réponse (voir le Tableau A.12).
0x11	IDLE 1 Pas de demande de transmission de ISDU.
0x12	IDLE 2: Réservé pour usage ultérieur Pas de demande de transmission de ISDU.
0x13 à 0x1E	Réservé)
0x1F	ABORT Abandon de l'ensemble du service. Le Maître répond en rejetant les données de réponse reçues. Le Dispositif répond en rejetant les données de demande reçues et peut générer un abandon.

Dans l'état Idle_1, les valeurs 0x12 à 0x1F ne doivent pas donner lieu à une erreur de communication.

7.3.6.3 Diagramme d’états du module de traitement des ISDU du Maître

La Figure 49 illustre le diagramme d’états du module de traitement des ISDU du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation
Data written	Données écrites
Error	Erreur
Transmission completed	Transmission achevée

Figure 49 – Diagramme d'états du module de traitement des ISDU du Maître

Le Tableau 51 présente les tables de transitions d'états du module de traitement des ISDU du Maître.

Tableau 51 – Tables de transitions d'états du module de traitement des ISDU du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
Idle_1	Attente de transmission des prochaines Données à la Demande
ISDURequest_2	Transmission des données de demande d'ISDU
ISDUWait_3	Attente de réponse du Dispositif. Respect de ISDUTime
ISDUError_4	Gestion des erreurs après détection: Invocation de la réponse DL_ISDU_Transport négative avec ISDUTransportErrorInfo
ISDUResponse_5	Obtention des données de réponse du Dispositif

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	2	Invocation de OD.req avec condition de début d'écriture des ISDU: OD.req (Écriture, ISDU, flowCtrl = DÉBUT, données)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

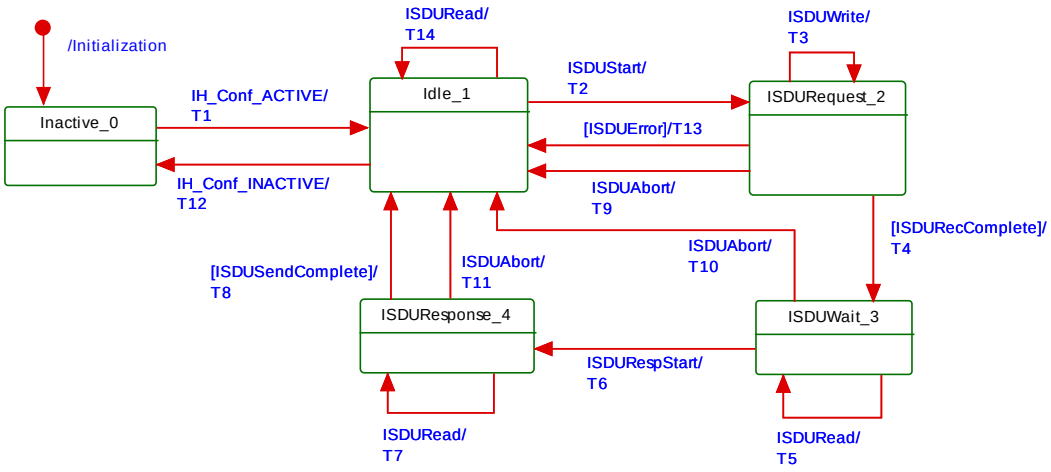
TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T3	2	2	Invocation de OD.req avec écriture de données ISDU et FlowCTRL dans les conditions définies au Tableau 50
T4	2	3	Démarrage du temporisateur (ISDUTime)
T5	3	3	Invocation de OD.req avec condition de début de lecture des ISDU: OD.req (Lecture, ISDU, flowCtrl = DÉBUT)
T6	3	5	Arrêt du temporisateur (ISDUTime)
T7	5	5	Invocation de OD.req avec lecture de données ISDU et FlowCTRL dans les conditions définies au Tableau 50
T8	5	1	Invocation de confirmation positive de DL_ ISDUTransport
T9	3	4	-
T10	5	4	-
T11	4	1	Invocation de OD.req avec abandon des ISDU: OD.req (Lecture, ISDU, flowCtrl = ABANDON) Invocation de confirmation négative de DL_ ISDUTransport
T12	2	4	-
T13	1	1	Invocation de OD.req avec des données appropriées Invocation de confirmation positive de DL_ReadParam/DL_WriteParam
T14	1	1	Invocation de OD.req avec message au repos: OD.req (Lecture, ISDU, flowCtrl = REPOS)
T15	4	1	En cas de perte de communication, le module de traitement des messages informe le module de traitement de Mode DL qui à son tour utilise l'invocation réglementaire IH_Conf_INACTIVE. Aucune action n'est requise au cours de cette transition.
T16	1	0	-
T17	3	4	-
T18	5	4	-
T19	2	4	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
ISDUTime	Temps	Mesure du temps de réponse du Dispositif (chien de garde, voir le Tableau 97)
ResponseStart	Service	OD.cnf (données autres que ISDU_BUSY)
ParamRequest	Service	DL_ReadParam ou DL_WriteParam
Erreur	Variable	Toute erreur détectable dans la transmission de ISDU ou demandes DL_ISDUAbort , ou toute violation du délai d'acquittement des ISDU (voir le Tableau 97)

7.3.6.4 Diagramme d'états du module de traitement des ISDU du Dispositif

La Figure 50 illustre le diagramme d'états du module de traitement des ISDU du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation

Figure 50 – Diagramme d'états du module de traitement des ISDU du Dispositif

Le Tableau 52 présente les tables de transitions d'états du module de traitement des ISDU du Dispositif.

Tableau 52 – Tables de transitions d'états du module de traitement des ISDU du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
Idle_1	Attente de transmission des ISDU suivantes
ISDURequest_2	Réception de demande d'ISDU
ISDUWait_3	Attente de données à transmettre en provenance de la couche d'application (voir DL_ISDUTransport)
ISDUResponse_4	Transmission des données de réponse d'ISDU

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	2	Début de réception des données de demande d'ISDU
T3	2	2	Réception des données de demande d'ISDU
T4	2	3	Invocation de DL_ ISDUTransport.ind vers la couche AL (voir 7.2.1.6)
T5	3	3	Invocation de OD.rsp avec l'indication "occupé" (voir le Tableau A.14)
T6	3	4	-
T7	4	4	Invocation de OD.rsp avec données de réponse d'ISDU
T8	4	1	-
T9	2	1	-
T10	3	1	Invocation de DL_ ISDUAbort
T11	4	1	Invocation de DL_ ISDUAbort
T12	1	0	-

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T13	2	1	-
T14	1	1	Invocation de OD.rsp avec l'indication "Pas de Service" (voir le Tableau A.12 et le Tableau A.14)

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
ISDUStart	Service	OD.ind(Écriture, ISDU, Début, Données)
ISDUWrite	Service	OD.ind(Écriture, ISDU, FlowCtrl, Données)
ISDURecComplete	Protection	Si OD.ind(Lecture, ISDU, Début, ...) reçue
ISDURespStart	Service	DL_ ISDUTransport.rsp()
ISDURead	Service	OD.ind(Lecture, ISDU, Début ou FlowCtrl, ...)
ISDUSendComplete	Protection	Si OD.ind(Lecture, ISDU, REPOS, ...) reçue
ISDUAbort	Service	OD.ind(Lecture/Écriture, ISDU, Abandon, ...)
ISDUError	Protection	Si la structure des ISDU est incorrecte

7.3.7 Module de traitement des commandes

7.3.7.1 Généralités

Le module de traitement des commandes passe le code de contrôle (PDOUTVALID ou PDOUTINVALID) contenu dans la primitive de service DL_Control.req au module de traitement des messages à fonctionnement cyclique, via le service OD.req et MasterCommands. Le module de traitement des messages utilise le canal de communication de page.

Les codes de contrôle admissibles pour les Données de Processus de sortie sont énumérés dans le Tableau 53.

Tableau 53 – Codes de contrôle

Code de contrôle	MasterCommand	Description
PDOUTVALID	ProcessDataOutputOperate	Données de Processus de sortie valides
PDOUTINVALID	DeviceOperate	Données de Processus de sortie non valides ou manquantes

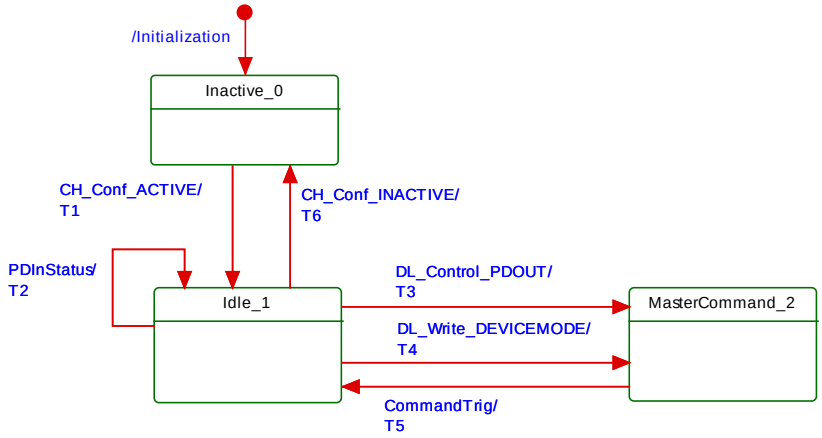
Le module de traitement des commandes reçoit des informations d'état des Données de Processus d'entrée, via le service PDInStatus et les propage dans une primitive de service DL_Control.ind.

En outre, le module de traitement des commandes traduit les demandes de modification de mode du Dispositif provenant de la gestion système en des commandes MasterCommands correspondantes (voir le Tableau B.2).

7.3.7.2 Diagramme d'états du module de traitement des commandes du Maître

La Figure 51 illustre le diagramme d'états du module de traitement des commandes du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation

Figure 51 – Diagramme d'états du module de traitement des commandes du Maître

Le Tableau 54 présente les tables de transitions d'états du module de traitement des commandes du Maître.

Tableau 54 – Tables de transitions d'états du module de traitement des commandes du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation par le module de traitement de mode DL
Idle_1	Attente de nouvelle commande en provenance de la couche AL: DL_Control (état des PD de sortie) ou en provenance de la SM: DL_Write (changement de mode du Dispositif, par exemple en OPERATE), ou attente de la primitive de service PDInStatus.ind.
MasterCommand_2	Préparation de données pour la primitive de service OD.req . Attente de demande de la part du module de traitement des OD (CommandTrig).

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	1	Si le service PDInStatus.ind = VALID, invocation de DL_Control.ind (VALID) pour signaler des Données de Processus d'entrée valides vers AL. Si le service PDInStatus.ind = INVALID, invocation de DL_Control.ind (INVALID) pour signaler des Données de Processus d'entrée non valides vers AL.
T3	1	1	Si le service DL_Control.req = PDOUTVALID, invocation de OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x98). Si le service DL_Control.req = PDOUTINVALID, invocation de OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x99). Voir le Tableau B.2.
T4	1	2	Les services DL_Write_DEVICEMODE traduisent en: INACTIVE: OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x5A) STARTUP: OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x97) PREOPERATE: OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x9A) OPERATE: OD.req (ÉCRITURE, PAGE, 0, 1, MasterCommand = 0x99)
T5	2	1	Une invocation CommandTrig à partir du module de traitement des OD force le module de traitement des commandes à invoquer la primitive de service OD.req et par conséquent le module de traitement des messages à envoyer au Dispositif la MasterCommand appropriée.

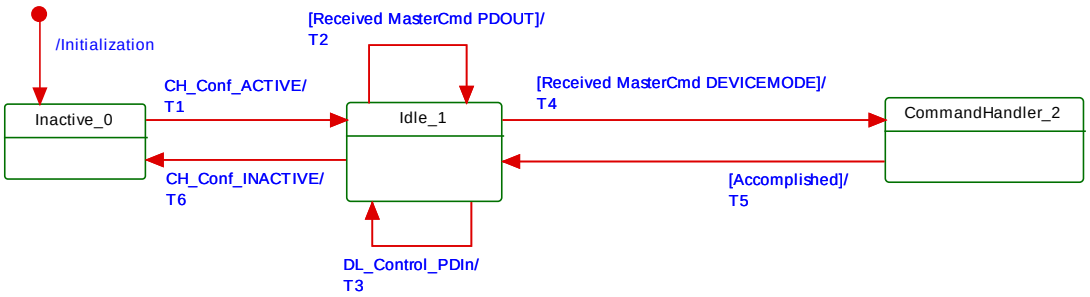
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T6	1	0	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
DEVICEMODE	Étiquette	N'importe quel mode du Dispositif: INACTIVE, STARTUP, PREOPERATE, ou OPERATE
PDout	Étiquette	Un des deux codes de contrôle de sortie: PDOUTVALID ou PDOUTINVALID (voir le Tableau 53)

7.3.7.3 Diagramme d'états du module de traitement des commandes du Dispositif

La Figure 52 illustre le diagramme d'états du module de traitement des commandes du Dispositif. Il est principalement piloté par des MasterCommands provenant du module de traitement des commandes du Maître pour contrôler les modes du Dispositif et l'état des Données de Processus de sortie. Il contrôle également l'état des Données de Processus d'entrée via le service PDInStatus.



Légende

Anglais	Français
Initialization	Initialisation
Received MasterCmd DEVICEMODE	Commande Maître DEVICEMODE reçue
Accomplished	Exécutée

Figure 52 – Diagramme d'états du module de traitement des commandes du Dispositif

Le Tableau 55 présente les tables de transitions d'états du module de traitement des commandes du Dispositif.

Tableau 55 – Tables de transitions d'états du module de traitement des commandes du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
Idle_1	Attente de la MasterCommand suivante
CommandHandler_2	Décomposition de la MasterCommand et invocation d'actions spécifiques (voir B.1.2): Si MasterCommand = 0x5A, passage du Dispositif à l'état INACTIVE. Si MasterCommand = 0x97, passage du Dispositif à l'état STARTUP. Si MasterCommand = 0x9A, passage du Dispositif à l'état PREOPERATE. Si MasterCommand = 0x99, passage du Dispositif à l'état OPERATE.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	1	Invocation de DL_Control.ind (PDOUTVALID) si MasterCommand reçue = 0x98. Invocation de DL_Control.ind (PDOUTINVALID) si MasterCommand reçue = 0x99.
T3	1	1	Si service DL_Control.req (VALID), invocation de PDInStatus.req (VALID). Si service DL_Control.req (INVALID), invocation de PDInStatus.req (INVALID). Le module de traitement des messages utilise le service PDInStatus pour initialiser/réinitialiser le drapeau d'état des PD (voir A.1.5)
T4	1	2	-
T5	2	1	-
T6	1	0	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
<Aucun>		

7.3.8 Module de traitement d'Événements

7.3.8.1 Événements

Il existe deux types d'Événements, les uns détaillés et les autres non détaillés. Des événements non détaillés peuvent avoir été utilisés dans des Dispositifs patrimoniaux, mais ils ne doivent pas être utilisés pour des Dispositifs conformes à la présente norme. Cependant, tous les Maîtres doivent prendre en charge le traitement des deux types d'Événements, détaillés et non détaillés.

La structure générale et le codage des Événements sont définis en A.6. Les codes d'événements non détaillés sont donnés dans le Tableau A.16. Les codes d'événements détaillés sont spécifiés en Annexe D. La structure de la mémoire d'Événements pour des codes d'événements détaillés dans un Dispositif donné est définie dans le Tableau 56.

Tableau 56 – Mémoire d'Événements

Adresse	Numéro d'emplacement de l'événe-ment	Nom du paramètre	Description
0x00		StatusCode	Récapitulatif des informations d'état et d'erreurs. Également utilisé pour contrôler l'accès en lecture pour des messages individuels.
0x01	1	EventQualifier 1	Type, mode et origine de l'Événement
0x02		EventCode 1	Code d'Événement sur 16 bits de l'Événement
0x03			
0x04	2	EventQualifier 2	Type, mode et origine de l'Événement
0x05		EventCode 2	Code d'Événement sur 16 bits de l'Événement
0x06			
...			
0x10	6	EventQualifier 6	Type, mode et origine de l'Événement
0x11		EventCode 6	Code d'Événement sur 16 bits de l'Événement
0x12			
0x13 à 0x1F			Réservé pour un usage futur

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

7.3.8.2 Traitement des Événements

La couche AL du Dispositif écrit un Événement dans la mémoire d'Événements, puis établit le bit "drapeau d'Événement" qui est ensuite envoyé au Maître avec le message suivant dans l'octet CKS (voir 7.3.3.2 et A.1.5).

À la réception d'un message de réponse du Dispositif avec le bit "drapeau d'Événement" = 1, le Maître doit faire passer le module de traitement des ISDU en module de traitement d'Événements. Le module de traitement d'Événements commence à lire le StatusCode.

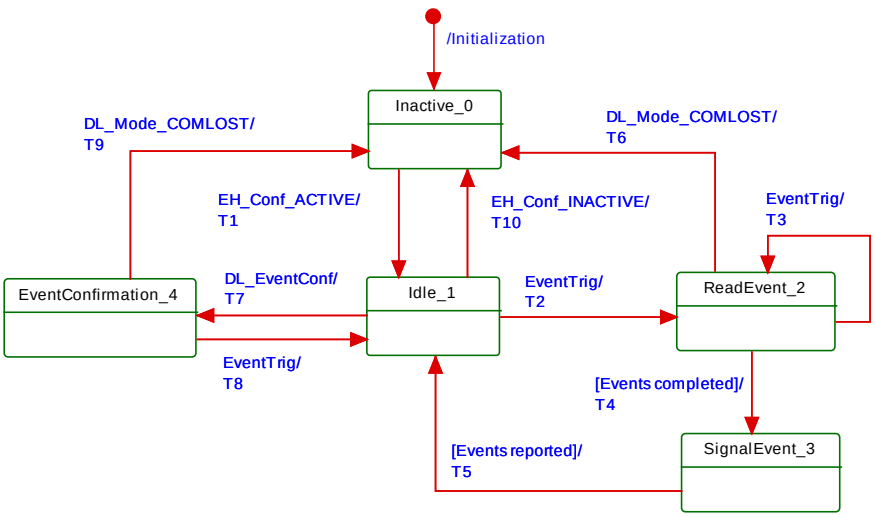
Si le bit "Détails de l'Événement" est établi (voir la Figure A.23), le Maître doit lire les détails des Événements indiqués dans le StatusCode à partir de la mémoire d'Événements. Après avoir lu les détails d'un Événement, il doit invoquer le service DL_Event.ind. Après réception du service DL_EventConf, le Maître doit écrire dans le StatusCode les éventuelles données nécessaires pour réinitialiser le bit "drapeau d'Événement". La gestion des événements du côté Maître doit être menée à terme quel que soit le contenu des données d'Événements reçues (EventQualifier, EventCode).

Si le bit "Détails d'Evénement" n'est pas établi (voir la Figure A.22), le module de traitement d'Événements du Maître doit générer les Événements normalisés conformément au Tableau A.16 commençant par le bit de poids fort dans le EventCode.

L'accès en écriture au StatusCode indique au Dispositif la fin du traitement des Événements. Le Dispositif doit ignorer les données de cet accès en écriture du Maître. Le Dispositif réinitialise alors le bit "Drapeau d'Événement" et peut désormais modifier le contenu des champs de la mémoire d'Événements.

7.3.8.3 Diagramme d'états du module de traitement d'Événements du Maître

La Figure 53 illustre le diagramme d'états du module de traitement d'Événements du Maître.



Légende

Anglais	Français
Initialization	Initialisation
Events reported	Événements rapportés

Figure 53 – Diagramme d'états du module de traitement d'Événements du Maître

Le Tableau 57 présente les tables de transitions d'états du module de traitement d'Événements du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 57 – Tables de transitions d'états du module de traitement d'Événements du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
Idle_1	Attente de l'indication d'Événement suivante ("EventTrig" par l'intermédiaire du module de traitement de Données à la Demande) ou confirmation d'Événement par l'intermédiaire du service DL_EventConf en provenance de la couche AL du Maître.
ReadEvent_2	Lecture, message par message, du jeu de données d'Événement du Dispositif via l'adresse de la mémoire d'Événements. Vérification du Code d'État pour un certain nombre d'Événements activés (voir le Tableau 56).
SignalEvent_3	Analyse des données d'Événement et invocation de l'indication DL_Event vers la couche AL du Maître (voir 7.2.1.15) pour chaque Événement disponible.
EventConfirmation_4	Attente de la transmission de la confirmation d'Événement via le service OD.req vers le Dispositif

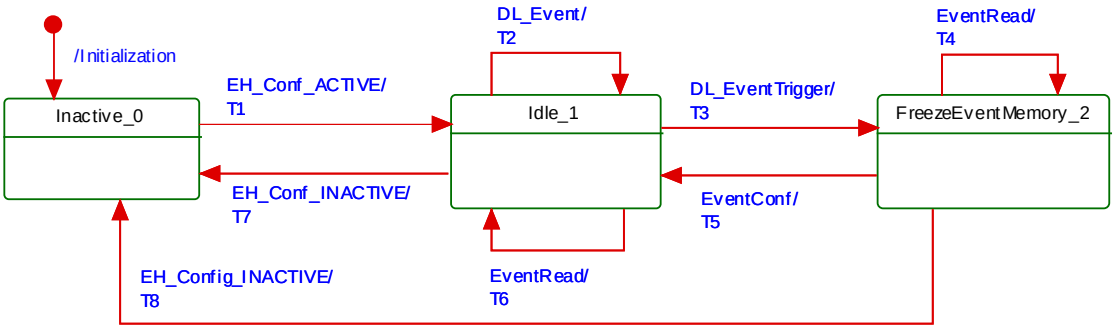
TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	2	Lecture de l'octet de Code d'État d'Événement via le service OD.req (Lecture, DIAGNOSTIC, adresse de mémoire d'Événements = 0, 1)
T3	2	2	Lecture des octets en provenance de la mémoire d'Événements via le service OD.req (Lecture, DIAGNOSTIC, adresse de mémoire d'Événements incrémentée, 1)
T4	2	3	-
T5	3	1	-
T6	2	0	-
T7	1	4	-
T8	4	1	Invocation de OD.req (Écriture, DIAGNOSTIC, 0, 1, toutes données) avec accès en écriture au "Code d'État" (voir le Tableau 56) pour confirmation de la lecture de l'Événement au Dispositif
T9	4	0	-
T10	1	0	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
<Aucun>		

7.3.8.4 Diagramme d'états du module de traitement d'Événements du Dispositif

La Figure 54 illustre le diagramme d'états du module de traitement d'Événements du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation

Figure 54 – Diagramme d'états du module de traitement d'Événements du Dispositif

Le Tableau 58 présente les tables de transitions d'états du module de traitement d'Événements du Dispositif.

Tableau 58 – Tables de transitions d'états du module de traitement d'Événements du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
Idle_1	Attente du service DL-Event de la couche AL qui fournit les données d'Événement et du service DL_EventTrigger pour établir le bit "drapeau d'Événement " (voir A.1.5)
FreezeEventMemory_2	Attente de la lecture de la mémoire d'Événements et de la confirmation de lecture de la mémoire d'Événements par accès en écriture au Code d'État

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	1	Remplacement des entrées de la mémoire d'Événements par de nouvelles données d'Événement (voir le Tableau 56)
T3	1	2	Invocation du service EventFlag.req (Drapeau = VRAI) pour indiquer l'activation d'Événements au Maître via le bit "drapeau d'Événement". Marquage de tous les emplacements d'événements en mémoire comme non modifiables.
T4	2	2	Le Maître demande des données de la mémoire d'Événements via EventRead (= OD.ind). Envoi de données d'Événement par invocation de OD.rsp avec des données d'Événement de l'adresse de la mémoire des Événements demandés.
T5	2	1	Invocation du service EventFlag.req (Drapeau = FAUX) pour indiquer la désactivation d'Événements au Maître via le bit "drapeau d'Événement". Marquage de tous les emplacements d'événements en mémoire comme non valides selon A.6.3.
T6	1	1	Envoi du contenu de la mémoire d'Événements par invocation de OD.rsp avec des données d'Événement
T7	1	0	-
T8	2	0	Vidage des données de la mémoire d'Événements

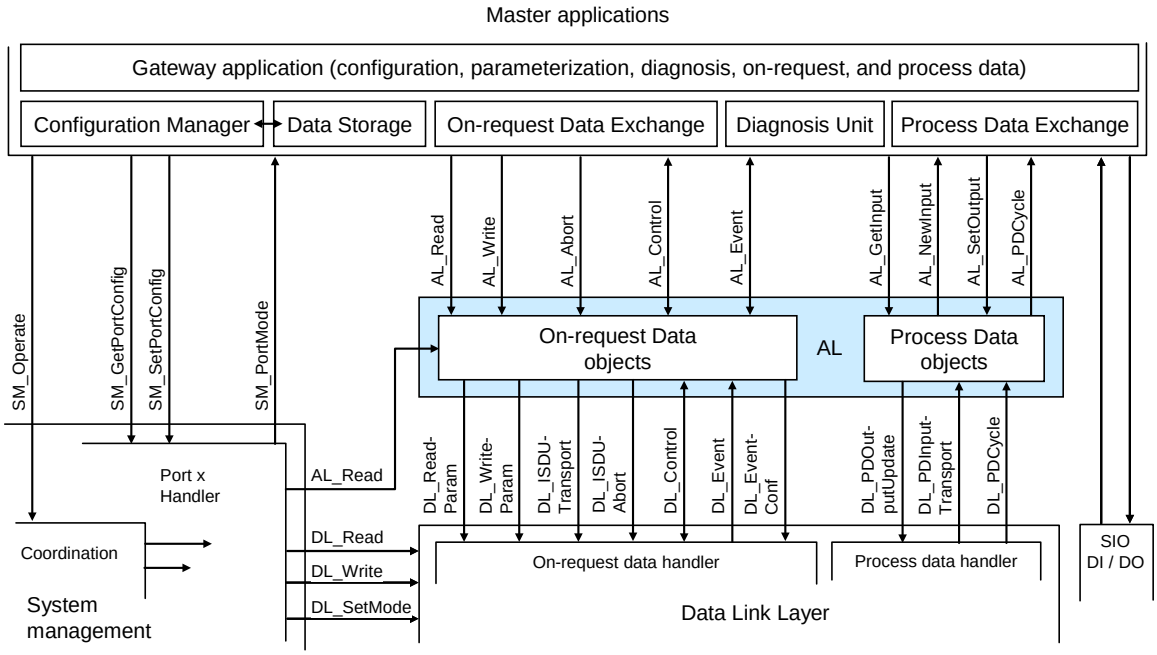
ÉLÉMENTS INTERNES	TYPE	DEFINITION
EventRead	Service	OD.ind (R, DIAGNOSIS, adresse de mémoire d'Événements, longueur, données)
EventConf	Service	OD.ind (W, DIAGNOSIS, adresse = 0, données = sans effet)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8 Couche d'application (AL)

8.1 Généralités

La Figure 55 illustre une vue globale de la structure et des services de la couche d'application (AL) du Maître.



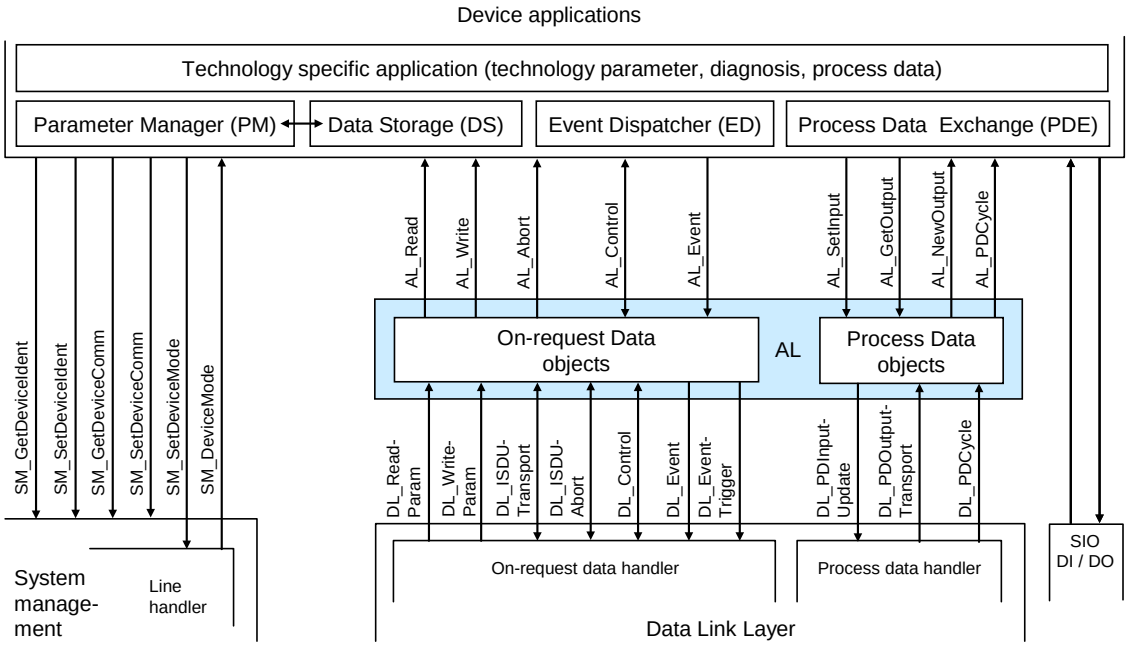
Légende

Anglais	Français
Master applications	Applications du Maître
Gateway application (configuration, parameterization, diagnosis, on-request and process data)	Application passerelle (configuration, paramétrage, diagnostic, données à la demande et de processus)
Configuration Manager	Gestionnaire de configuration
Data Storage	Stockage des données
On-request Exchange	Échange de données à la demande
Diagnosis Unit	Unité de Diagnostic
Process Data Exchange	Échange de Données de Processus
On-request Data objects	Objets de Données à la Demande
Process Data objects	Objets de Données de Processus
Port x Handler	Module de traitement de Port-x
Coordination	Coordination
System management	Gestion Système
On-request data handler	Module de traitement de Données à la Demande
Process data handler	Module de traitement des Données de Processus
Data Link Layer	Couche de liaison de données

Figure 55 – Structure et services de la couche d'application (Maître)

La Figure 56 illustre une vue globale de la structure et des services de la couche d'application (AL) du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device applications	Applications du Dispositif
Technology specific application (technology parameter, diagnosis, process data)	Application spécifique à la technologie (paramètres, diagnostic, données de processus liés à la technologie)
Parameter Manager (PM)	Gestionnaire de Paramètres (PM)
Data Storage (DS)	Stockage des Données (DS)
Event Dispatcher (ED)	Répartiteur d'Événements (ED)
Process Data Exchange (PDE)	Échange de Données de Processus (PDE)
On-request Data objects	Objets de Données à la Demande
Process Data objects	Objets de Données de Processus
System management	Gestion Système
Line handler	Module de traitement de ligne
On-request data handler	Module de traitement de Données à la Demande
Process data handler	Module de traitement des Données de Processus
Data Link Layer	Couche de liaison de données

Figure 56 – Structure et services de la couche d'application (Dispositif)

8.2 Services de la Couche d'Application

8.2.1 Services AL au sein du Maître et du Dispositif

L'Article 8 définit les services de la couche d'application (AL) à fournir au Maître, aux applications de Dispositif et à la gestion système via ses interfaces externes. Le Tableau 59 énumère les attributions du Maître et du Dispositif quant à leur rôle d'initiateur ou de destinataire des services individuels de couche d'application (AL). Des champs vides indiquent la non-disponibilité de ce service au niveau du Maître ou du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 59 – Services AL au sein du Maître et du Dispositif

Désignation du service	Maître	Dispositif
AL_Read	R	I
AL_Write	R	I
AL_Abort	R	I
AL_GetInput	R	
AL_NewInput	I	
AL_SetInput		R
AL_PDCycle	I	I
AL_GetOutput		R
AL_NewOutput		I
AL_SetOutput	R	
AL_Event	I / R	R
AL_Control	I / R	I / R
Légende (voir 3.3.4) I Initiateur du service R Destinataire (Répondeur) du service		

8.2.2 Services d'AL

8.2.2.1 AL_Read

Le service AL_Read est utilisé pour lire des Données à la Demande en provenance d'un Dispositif connecté à un port spécifique. Les paramètres des primitives de service sont énumérés dans le Tableau 60.

Tableau 60 – AL_Read

Nom du paramètre	.req	.ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Sous-index	M	M		
Résultat(+)			S	S(=)
Port				M
Données			M	M(=)
Résultat (-)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro de port pour les Données à la Demande à lire.

Type de paramètre: Unsigned8

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Index
Ce paramètre indique l'adresse des objets de Données à la Demande à lire provenant du Dispositif. L'Index 0 conjointement au Sous-index 0 couvre l'ensemble du jeu de Paramètres Directs de 0 à 15 (voir la page 1 de Paramètres Directs dans le Tableau B.1) ou, conjointement aux Sous-index 1 à 16, il couvre les paramètres individuels de 0 à 15. L'Index 1 conjointement au Sous-index 0 couvre l'ensemble du jeu de Paramètres Directs des adresses 16 à 31 (voir la page 2 de Paramètres Directs dans le Tableau B.1) ou, conjointement aux Sous-index 1 à 16, il couvre les paramètres individuels de 16 à 31. Il utilise le canal de communication de page (voir Figure 6) dans les deux cas et renvoie toujours un résultat positif. Le canal de communication des ISDU est utilisé pour tous les autres index (voir B.2).

Valeurs autorisées: 0 à 65 535 (Voir B.2.1 pour les contraintes)

Sous-index
Ce paramètre indique le numéro d'élément dans un objet structuré de Données à la Demande. Une valeur 0 indique l'ensemble du jeu d'éléments.

Valeurs autorisées: 0 à 255

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Port
Ce paramètre contient le numéro de port des Données à la Demande requises.

Données
Ce paramètre contient les valeurs lues des Données à la Demande.

Type de paramètre: Chaîne d'octets

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Port
Ce paramètre contient le numéro de port pour les Données à la Demande requises.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées: voir l'Annexe C

NOTE La couche AL met en correspondance les ErrorInfos de couche DL avec ses propres ErrorInfos de couche AL en utilisant l'Annexe C.

8.2.2.2 AL_Write

Le service AL_Write est utilisé pour écrire des Données à la Demande vers un Dispositif connecté à un port spécifique. Les paramètres des primitives de service sont énumérés dans le Tableau 61.

Tableau 61 – AL_Write

Nom du paramètre	.req	.ind	.rsp	.cnf
Argument	M	M		
Port	M			
Index	M	M		
Sous-index	M	M		
Données	M	M(=)		
Résultat(+)			S	S(=)
Port				M

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Nom du paramètre	.req	.ind	.rsp	.cnf
Résultat (-)			S	S(=)
Port				M
ErrorInfo			M	M(=)

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro de port pour les Données à la Demande à écrire.

Type de paramètre: Unsigned8

Index
Ce paramètre indique l'adresse des objets de Données à la Demande à écrire vers le Dispositif. L'Index 0 renvoie toujours un résultat négatif. L'Index 1 conjointement au Sous-index 0 couvre l'ensemble du jeu de Paramètres Directs des adresses 16 à 31 (voir la page 2 de Paramètres Directs dans le Tableau B.1) ou, conjointement aux Sous-index 1 à 16, il couvre les paramètres individuels de 16 à 31. Il utilise le canal de communication de page (voir Figure 6) dans le cas de l'index 1 et renvoie toujours un résultat positif. Le canal de communication des ISDU est utilisé pour tous les autres index (voir B.2).

Valeurs autorisées: 1 à 65 535 (voir le Tableau 97)

Sous-index
Ce paramètre indique le numéro d'élément dans un objet structuré de Données à la Demande. Une valeur 0 indique l'ensemble du jeu d'éléments.

Valeurs autorisées: 0 à 255

Données
Ce paramètre contient les valeurs des Données à la Demande.

Type de paramètre: Chaîne d'octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Port
Ce paramètre contient le numéro de port des Données à la Demande.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Port
Ce paramètre contient le numéro de port des Données à la Demande.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées: voir l'Annexe C

8.2.2.3 AL_Abort

Le service AL_Abort est utilisé pour abandonner un service AL_Read ou AL_Write en cours sur un port spécifique. L'invocation de ce service abandonne la réponse à un service AL_Read ou AL_Write en cours au niveau du Maître. Les paramètres des primitives de service sont énumérés dans le Tableau 62.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 62 – AL_Abort

Nom du paramètre	.req	.ind
Argument	M	M
Port	M	

Argument
Le paramètre spécifique au service est transmis dans l'argument.

Port
Ce paramètre contient le numéro de port du service à abandonner.

8.2.2.4 AL_GetInput

Le service AL_GetInput lit les données d'entrée dans les Données de Processus fournies par la couche de liaison de données d'un Dispositif connecté à un port spécifique. Les paramètres des primitives de service sont énumérés dans le Tableau 63.

Tableau 63 – AL_GetInput

Nom du paramètre	.req	.cnf
Argument	M	
Port	M	
Résultat(+)		S
Port		M
InputData		M
Résultat (-)		S
Port		M
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro de port pour les Données de Processus à lire.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Port
Ce paramètre contient le numéro de port des Données de Processus.

InputData
Ce paramètre contient les valeurs des Données de Processus d'entrée demandées du port spécifié.

Type de paramètre: Chaîne d'octets

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Port
Ce paramètre contient le numéro de port des Données de Processus.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_DATA (La DL n'a pas fourni de Données de Processus)

8.2.2.5 AL_NewInput

Le service local AL_NewInput indique la réception de données d'entrée mises à jour dans les Données de Processus d'un Dispositif connecté à un port spécifique. Les paramètres des primitives de service sont énumérés dans le Tableau 64.

Tableau 64 – AL_NewInput

Nom du paramètre	.ind
Argument	M
Port	M

Argument
Le paramètre spécifique au service est transmis dans l'argument.

Port
Ce paramètre spécifie le numéro de port des Données de Processus reçues.

8.2.2.6 AL_SetInput

Le service local AL_SetInput met à jour les données d'entrée dans les Données de Processus d'un Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 65.

Tableau 65 – AL_SetInput

Nom du paramètre	.req	.cnf
Argument	M	
InputData	M	
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

InputData
Ce paramètre contient les valeurs de Données de Processus des données d'entrée à transmettre.

Type de paramètre: Chaîne d'octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (Service non disponible en l'état courant)

8.2.2.7 AL_PDCycle

Le service local AL_PDCycle indique la fin d'un cycle de Données de Processus. L'application du Dispositif peut utiliser ce service pour transmettre de nouvelles données d'entrée à la couche d'application via AL_SetInput. Les paramètres des primitives de service sont énumérés dans le Tableau 66.

Tableau 66 – AL_PDCycle

Nom du paramètre	.ind
Argument	
Port	O

Argument
Le paramètre spécifique au service est transmis dans l'argument.

Port
Ce paramètre contient le numéro de port des nouvelles Données de Processus reçues (Maître uniquement).

8.2.2.8 AL_GetOutput

Le service AL_GetOutput lit les données de sortie dans les Données de Processus fournies par la couche de liaison de données du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 67.

Tableau 67 – AL_GetOutput

Nom du paramètre	.req	.cnf
Argument	M	
Résultat(+)		S
OutputData		M
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

OutputData
Ce paramètre contient les valeurs de Données de Processus des données de sortie demandées.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Type de paramètre: Chaîne d'octets

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
NO_DATA (La DL n'a pas fourni de Données de Processus)

8.2.2.9 AL_NewOutput

Le service local AL_NewOutput indique la réception de données de sortie mises à jour dans les Données de Processus d'un Dispositif. Ce service n'a pas de paramètres. Les primitives de service sont présentées dans le Tableau 68.

Tableau 68 – AL_NewOutput

Nom du paramètre	.ind
<Aucun>	

8.2.2.10 AL_SetOutput

Le service local AL_SetOutput met à jour les données de sortie dans les Données de Processus d'un Maître. Les paramètres des primitives de service sont énumérés dans le Tableau 69.

Tableau 69 – AL_SetOutput

Nom du paramètre	.req	.cnf
Argument	M	
Port	M	
OutputData	M	
Résultat(+)		S
Port		M
Résultat (-)		S
Port		M
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro de port des Données de Processus à écrire.

OutputData
Ce paramètre contient les données de sortie à écrire au port spécifié.

Type de paramètre: Chaîne d'octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Port
Ce paramètre contient le numéro de port des Données de Processus.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Port
Ce paramètre contient le numéro de port des Données de Processus.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (Service non disponible en l'état courant)

8.2.2.11 AL_Event

Le service AL_Event indique jusqu'à 6 états ou messages d'erreur en cours. L'origine d'un Événement peut être locale (Maître) ou distante (Dispositif). L'Événement peut être déclenché par une couche de communication ou par une application. Les paramètres des primitives de service sont énumérés dans le Tableau 70.

Tableau 70 – AL_Event

Nom du paramètre		.req	.ind	.rsp	.cnf
Argument		M	M	M	M
Port			M	M	M
EventCount		M	M		
Événement (1)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origine		M		
	EventCode	M	M		
...					
Événement (n)	Instance	M	M		
	Mode	M	M		
	Type	M	M		
	Origine		M		
	EventCode	M	M		

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro de port des données d'Événement.

EventCount
Ce paramètre indique le nombre n (1 à 6) d'Événements dans la mémoire d'Événements.

Event(x)
En fonction de EventCount, ce paramètre existe n fois. Chaque instance contient les éléments suivants:

Instance
Ce paramètre indique l'origine de l'Événement.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Valeurs autorisées: Application (voir le Tableau A.17)

Mode
Ce paramètre indique le mode de l'Événement.

Valeurs autorisées: SINGLESHOT, APPEARS, DISAPPEARS (voir le Tableau A.20)

Type
Ce paramètre indique la catégorie de l'Evénement.

Valeurs autorisées: ERROR, WARNING, NOTIFICATION (voir le Tableau A.19)

Origine
Ce paramètre indique si l'Événement a été généré dans la section de communication locale ou à distance (dans le Dispositif).

Valeurs autorisées: LOCAL, DISTANT

EventCode
Ce paramètre contient un code identifiant un certain Événement.

Valeurs autorisées: voir l'Annexe D

8.2.2.12 AL_Control

Le service AL_Control contient les informations d'état du qualificatif des Données de Processus transmises à et provenant de l'application du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 71.

Tableau 71 – AL_Control

Nom du paramètre	.req	.ind
Argument	M	M
Port	C	C
ControlCode	M	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Port
Ce paramètre contient le numéro du port concerné.

ControlCode
Ce paramètre contient l'état du qualificatif des Données de Processus (PD)

Valeurs autorisées:
VALID (Données de Processus d'entrée valides)
INVALID (Données de Processus d'entrée non valides)
PDOUTVALID (Données de Processus de sortie valides; voir le Tableau 53)
PDOUTINVALID (Données de Processus de sortie non valides, voir le Tableau 53)

8.3 Protocole de couche d'application

8.3.1 Aperçu

La Figure 7 montre que la couche d'application offre des services aux objets de données qui sont transformés en canaux de communication spéciaux de la couche de liaison de données.

La couche d'application gère le transfert de données avec tous les ports qui lui sont attribués. Ce qui signifie qu'il est nécessaire que les invocations de service AL identifient le port particulier auquel elles sont associées.

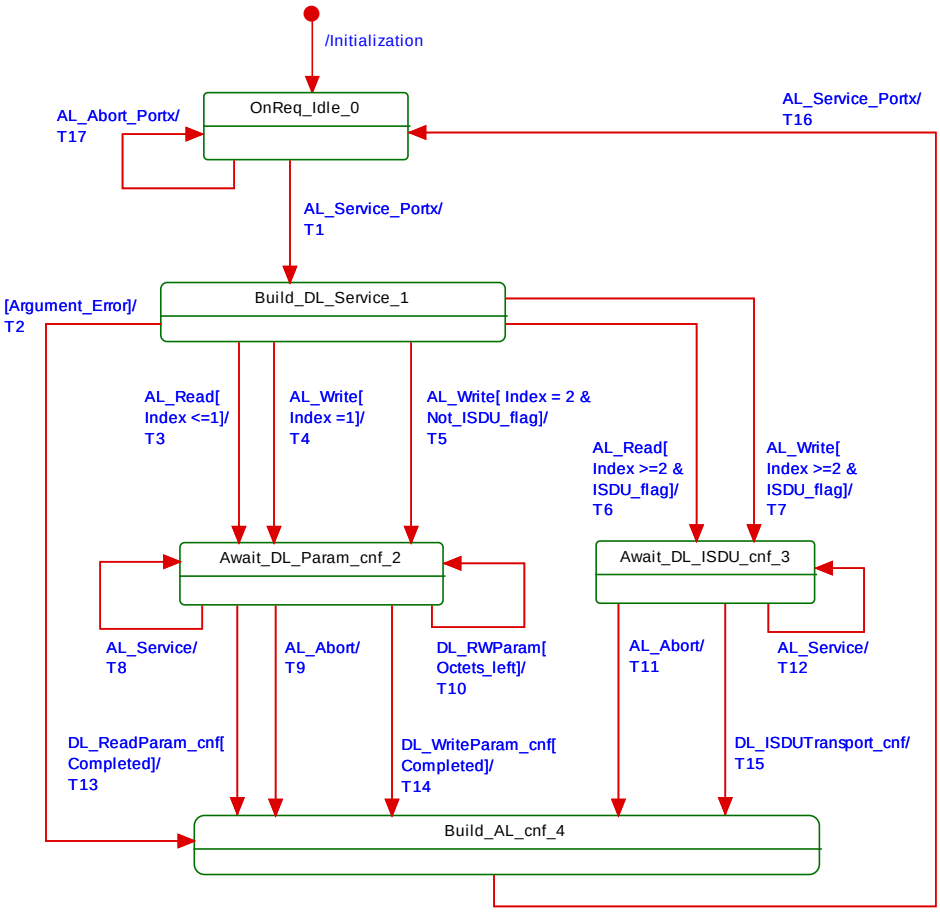
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.3.2 Transfert de Données à la Demande

8.3.2.1 Diagramme d'états d'OD de la couche AL du Maître

La Figure 57 illustre le diagramme d'états utilisé pour le traitement des Données à la Demande (OD) dans la couche d'application.

L'expression "AL_Service" se réfère à tout service AL du Tableau 59 et concernant les OD. "Portx" indique un numéro de port particulier.



Légende

Anglais	Français
Initialization	Initialisation

Figure 57 – Diagramme d'états d'OD de la couche AL du Maître

Le Tableau 72 présente les états et transitions pour le diagramme d'états d'OD de la couche AL du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 72 – États et transitions pour le diagramme d'états d'OD de la couche AL du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
OnReq_Idle_0	Les invocations de services AL provenant des applications du Maître ou du module de traitement de Portx de la SM (voir la Figure 55) peuvent être acceptées dans cet état.
Build_DL_Service_1	Dans cet état, les invocations de service AL sont vérifiées et les services DL correspondants sont générés dans les états ultérieurs. En cas d'erreur dans les arguments du service de couche AL, une confirmation AL négative est générée et renvoyée.
Await_DL_Param_cnf_2	Dans cet état, l'invocation de service AL est transformée en une séquence comportant autant d'invocations DL_ReadParam ou DL_WriteParam que nécessaire (accès à la page de Paramètres Directs; voir le canal de communication de page en Figure 6). Toutes les invocations de service AL en asynchrone, à l'exception de AL_Abort sont rejetées (voir 3.3.7).
Await_DL_ISDU_cnf_3	Dans cet état, l'invocation de service AL est transformée en une invocation de service DL_ISDUTransport (voir canal de communication des ISDU en Figure 6). Toutes les invocations de service AL en asynchrone, à l'exception de AL_Abort sont rejetées (voir 3.3.7).
Build_AL_cnf_4	Dans cet état, une confirmation de service AL est générée en fonction d'une erreur d'argument, de la confirmation de service DL ou d'un service AL_Abort.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	Mémorisation du numéro de port "Portx".
T2	1	4	Préparation d'une confirmation de service AL négative.
T3	1	2	Préparation de DL_ReadParam pour l'Index 0 ou 1.
T4	1	2	Préparation de DL_WriteParam pour l'Index 1.
T5	1	2	Préparation de DL_WriteParam pour l'Index 2 si le Dispositif ne prend pas en charge les ISDU.
T6	1	3	Préparation de DL_ ISDUTransport(read)
T7	1	3	Préparation de DL_ ISDUTransport(write)
T8	2	2	Renvoi d'une confirmation de service AL négative sur cette invocation de service asynchrone.
T9	2	4	Toutes les actions courantes de service DL sont abandonnées et une confirmation de service AL négative est préparée.
T10	2	2	Invocation du service DL_ReadParam ou DL_WriteParam suivant si les OD n'ont pas toutes été transférées.
T11	3	4	Toutes les actions courantes de service DL sont abandonnées et une confirmation de service AL négative est préparée.
T12	3	3	Renvoi d'une confirmation de service AL négative sur cette invocation de service asynchrone.
T13	2	4	Préparation d'une confirmation de service AL positive.
T14	2	4	Préparation d'une confirmation de service AL positive.
T15	3	4	Préparation d'une confirmation de service AL positive.
T16	4	0	Renvoi d'une confirmation de service AL positive avec le numéro de port "Portx".
T17	0	0	Renvoi d'une confirmation de service AL négative avec le numéro de port "Portx".

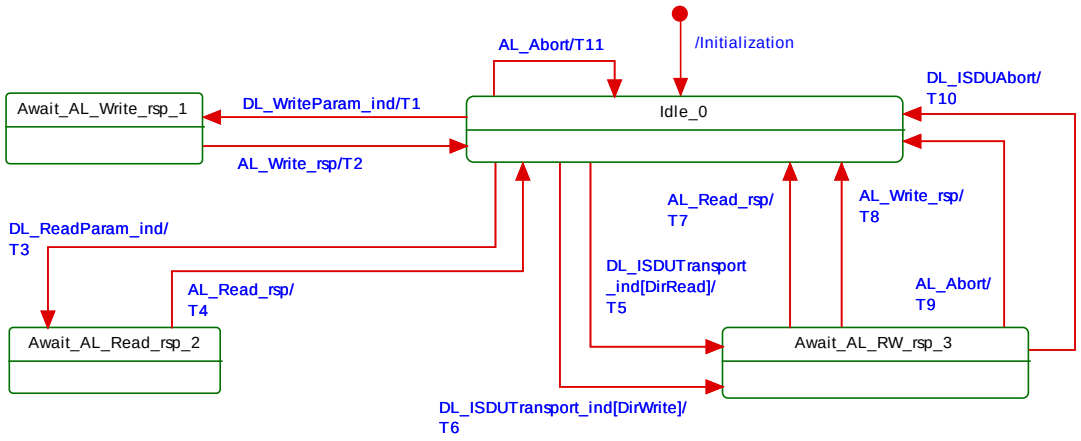
ÉLÉMENTS INTERNES	TYPE	DÉFINITION
Argument_Error	Bool	Valeurs illicites dans le corps du service, par exemple "Numéro de port ou Index hors limites"
DL_RWParam	Étiquette	"DL_RWParam": DL_WriteParam_cnf ou DL_ReadParam_cnf

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
Completed	Bool	Il ne reste plus d'autres OD à transférer
Octets_left	Bool	Il reste d'autres OD à transférer
Portx	Variable	Variable du corps du service indiquant le numéro de port
ISDU_Flag	Bool	Le Dispositif prend en charge les ISDU
AL_Service	Étiquette	L'expression "AL_Service" se réfère à tout service AL du Tableau 59 et concernant les OD.

8.3.2.2 Diagramme d'états d'OD de la couche AL du Dispositif

La Figure 58 illustre le diagramme d'états utilisé pour le traitement des Données à la Demande (OD) dans la couche d'application d'un Dispositif.



Légende

Anglais	Français
Initialization	Initialisation

Figure 58 – Diagramme d'états d'OD de la couche AL du Dispositif

Le Tableau 73 présente les états et transitions pour le diagramme d'états d'OD de la couche AL du Dispositif.

Tableau 73 – États et transitions pour le diagramme d'états d'OD de la couche AL du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Idle_0	La couche AL du Dispositif est en attente d'invocations de service DL subordonnées déclenchées par des messages du Maître.
Await_AL_Write_rsp_1	La couche AL du Dispositif est en attente d'une réponse de l'application spécifique à la technologie (accès en écriture à la page de Paramètres Directs).
Await_AL_Read_rsp_2	La couche AL du Dispositif est en attente d'une réponse de l'application spécifique à la technologie (accès en lecture à la page de Paramètres Directs).
Await_AL_RW_rsp_3	La couche AL du Dispositif est en attente d'une réponse de l'application spécifique à la technologie (accès en lecture ou en écriture via des ISDU).

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	Invocation de AL_Write.
T2	1	0	Invocation de DL_WriteParam (16 à 31).
T3	0	2	Invocation de AL_Read.
T4	2	0	Invocation de DL_ReadParam (0 à 31).
T5	0	3	Invocation de AL_Read.
T6	0	3	Invocation de AL_Write.
T7	3	0	Invocation de DL_ ISDUTransport(read)
T8	3	0	Invocation de DL_ ISDUTransport(write)
T9	3	0	AL_Read ou AL_Write en cours abandonné sur cette invocation de service asynchrone AL_Abort. Renvoi de DL_ ISDUTransport négatif (voir 3.3.7).
T10	3	0	Attente en cours de AL_Read ou AL_Write abandonnée.
T11	0	0	DL_ ISDUTransport courant abandonné. Toutes les OD sont mises à "0".

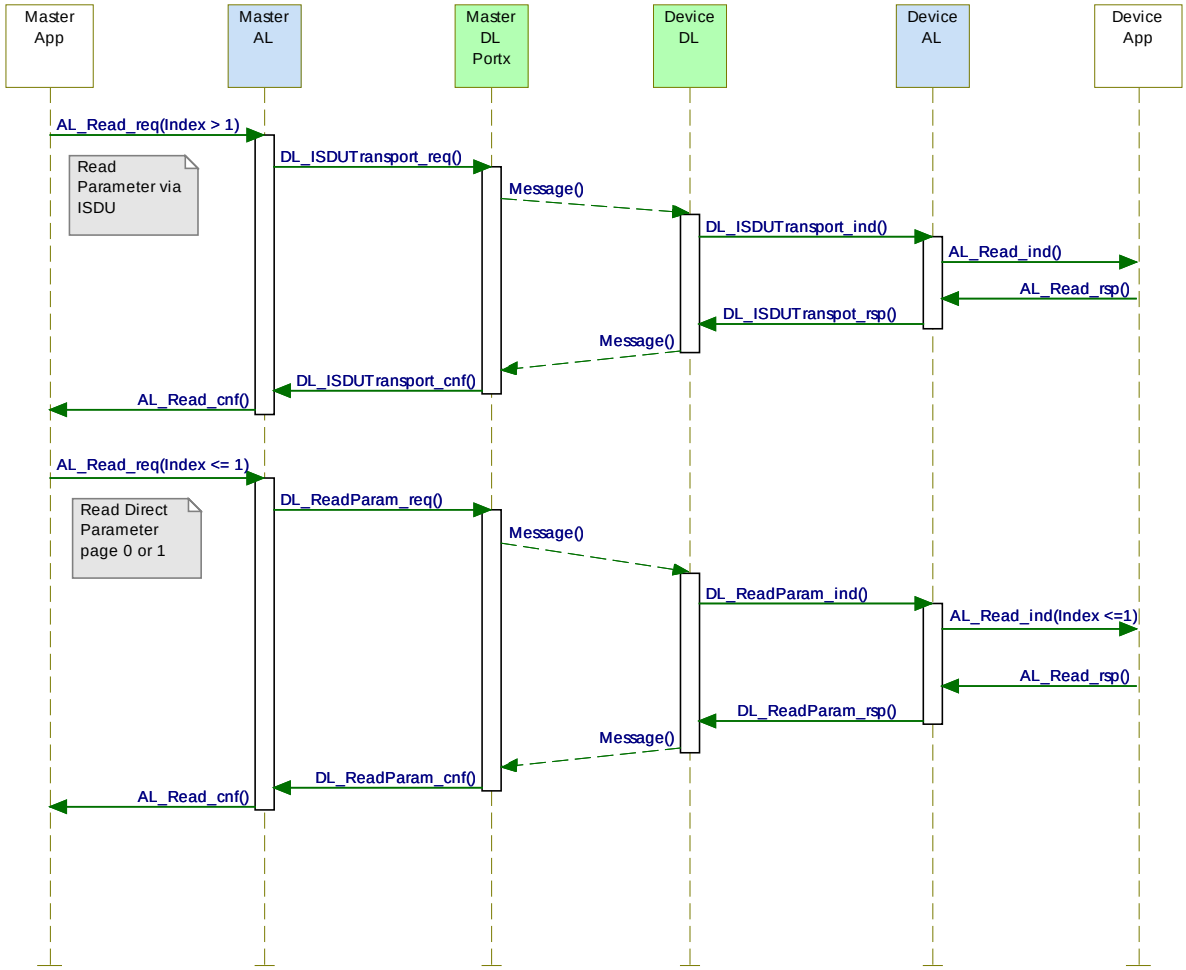
ÉLÉMENTS INTERNES	TYPE	DÉFINITION
DirRead	Bool	Sens d'accès: DL_ ISDUTransport(read) force un AL_Read
DirWrite	Bool	Sens d'accès: DL_ ISDUTransport(write) force un AL_Read

8.3.2.3 Diagrammes de séquence pour les Données à la Demande

Les Figure 59 à Figure 61 montrent l'ensemble des interactions entre le Maître et le Dispositif pour plusieurs cas d'utilisation de l'échange de Données à la Demande.

La Figure 59 illustre deux exemples d'échange de Données à la Demande. Pour des index > 1, cet échange est réalisé à l'aide d'unités ISDU et des services DL correspondants (canal de communication des ISDU, conformément à la Figure 6). Les accès aux pages 0 et 1 de Paramètres Directs utilisent des services DL différents (canal de communication de page, conformément à la Figure 6).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

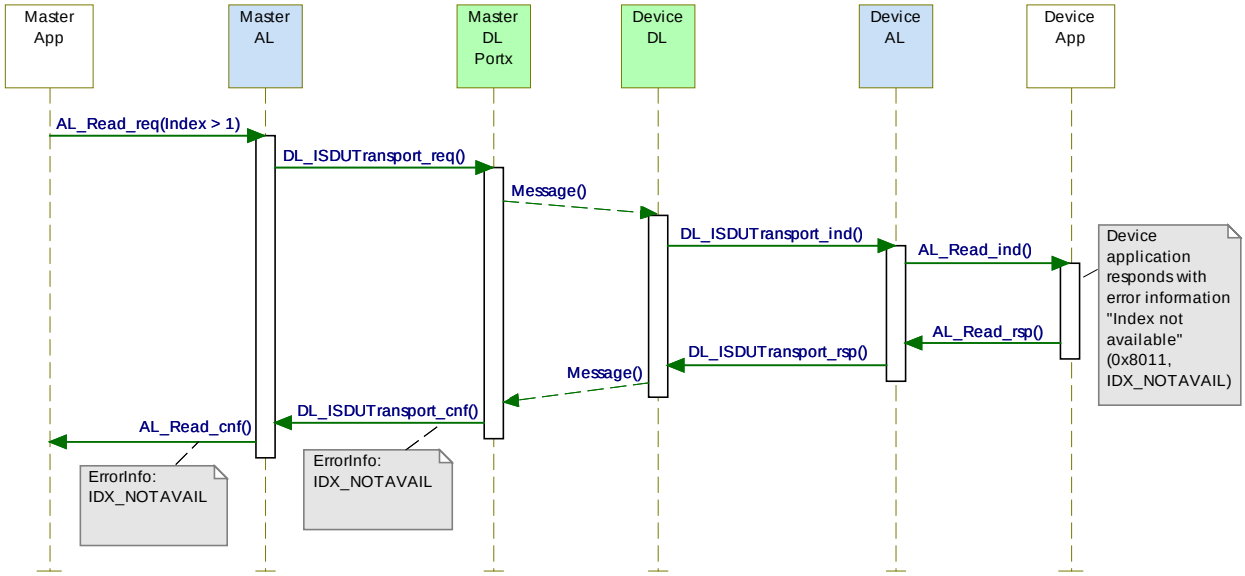
Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Read Parameter via ISDU	Lecture de paramètre via ISDU
Read Direct Parameter page 0 or 1	Lecture de page 0 ou 1 de Paramètres Directs

Figure 59 – Diagramme de séquence pour la transmission de Données à la Demande

La Figure 60 montre le comportement de l'échange de Données à la Demande en cas d'erreur, comme par exemple lorsque l'Index demandé n'est pas disponible (voir le Tableau C.1).

Une autre erreur possible a lieu lorsque l'application du Maître (passerelle) tente de lire un Index > 1 à partir d'un Dispositif qui ne prend pas en charge les ISDU. La couche AL du Maître répondrait immédiatement par "NO_ISDU_SUPPORTED" (PAS DE PRISE EN CHARGE D'ISDU) étant donné que les fonctionnalités du Dispositif sont acquises pendant le démarrage par lecture de la page 1 de Paramètres Directs via le paramètre "Capacité de séquence M " (voir le Tableau B.1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

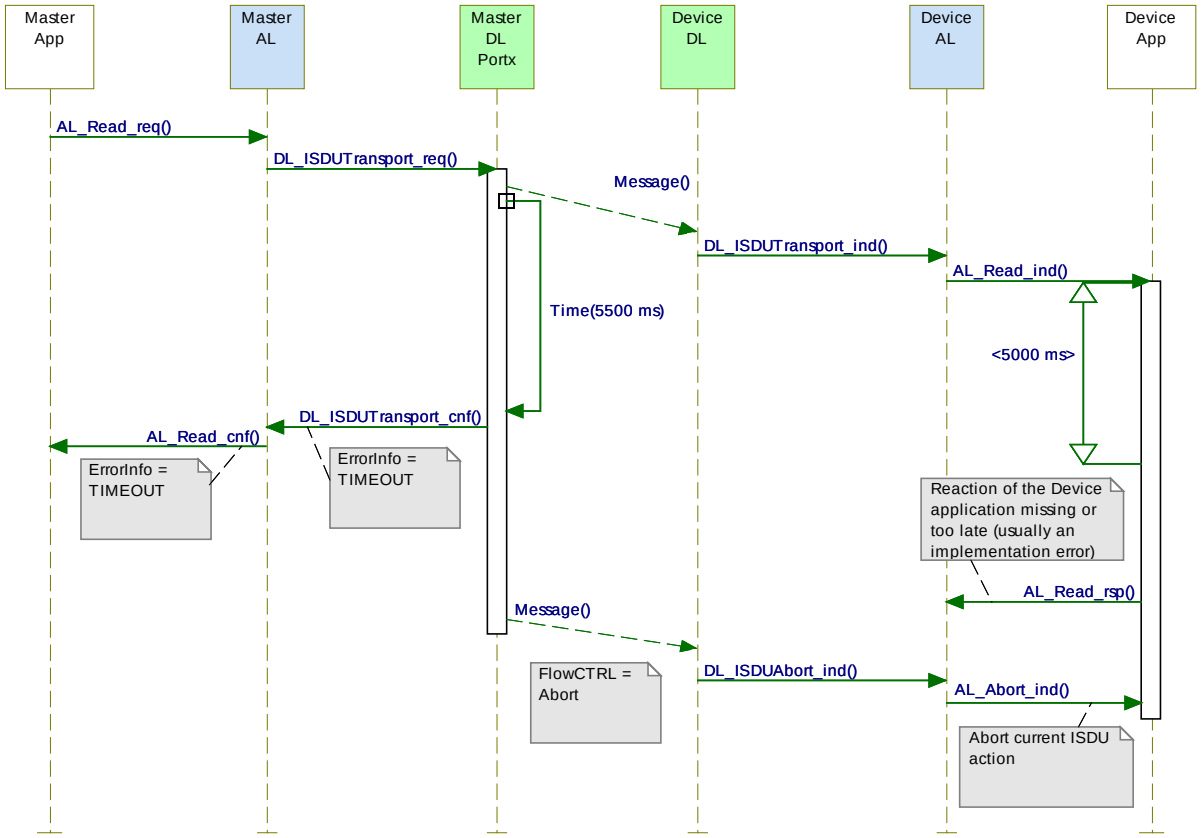
Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Device application responds with error information "Index not available" (0x8011, IDX_NOTAVAIL)	L'application du Dispositif répond avec l'information d'erreur "Index non disponible" (0x8011, IDX_NOTAVAIL)

Figure 60 – Diagramme de séquence pour des Données à la Demande en cas d'erreurs

La Figure 61 montre le comportement de l'échange de Données à la Demande en cas d'expiration du délai des ISDU (5 500 ms). Un Dispositif doit répondre dans un délai inférieur au "délai d'acquittement des ISDU" (voir 10.7.5).

NOTE Pour les constantes du système telles que le "délai d'acquittement des ISDU", voir le Tableau 97.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
ErrorInfo = TIMEOUT	ErrorInfo = Expiration du délai
Reaction of the Device application missing or too late (usually an implementation error)	Réaction de l'Application du Dispositif manquante ou trop tardive (généralement une erreur de mise en œuvre)
FlowCTRL = Abort	FlowCTRL = Abandon
Abort current ISDU action	abandon de l'action d'ISDU courante

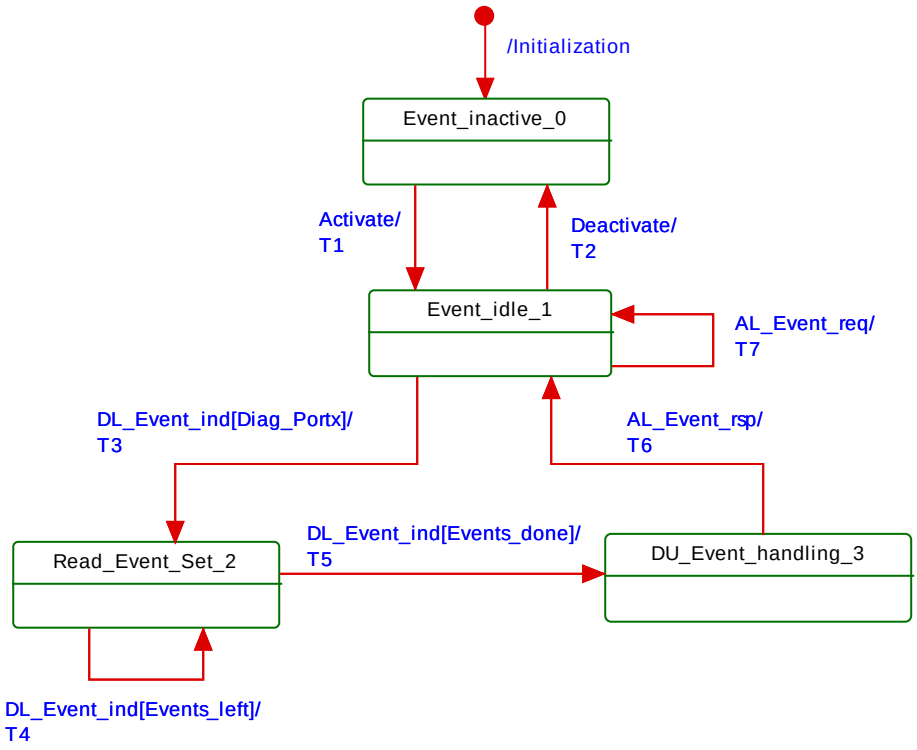
Figure 61 – Diagramme de séquence pour des Données à la Demande en cas d'expiration de délai

8.3.3 Traitement des Événements

8.3.3.1 Diagramme d'états d'Événements de la couche AL du Maître

La Figure 62 illustre le diagramme d'états d'Événements de la couche d'application du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation
Activate/	Activation/
Deactivate/	Désactivation/

Figure 62 – Diagramme d'états d'Événements de la couche AL du Maître

Le Tableau 74 définit les états et transitions du diagramme d'états d'Événements de la couche d'application du Maître.

Tableau 74 – États et transitions du diagramme d'états d'Événements de la couche AL du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Event_inactive_0	Le traitement d'Événements de couche AL du Maître est inactif.
Event_idle_1	La couche AL du Maître est prête à accepter des Événements DL (informations de diagnostic) à partir de la couche DL.
Read_Event_Set_2	La couche AL du Maître a reçu une DL_Event_ind avec des informations de diagnostic. Suite à cette première DL_Event.ind, la couche AL recueille l'ensemble du jeu (1 à 6) d'Événements DL du EventTrigger courant (voir 11.5).
DU_Event_handling_3	La couche AL du Maître reste dans cet état tant que l'Unité de Diagnostic (voir 11.5) n'a pas acquitté la AL_Event.ind.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	0	-
T3	1	2	-

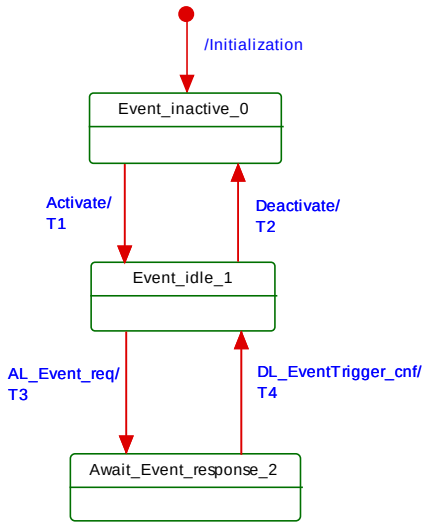
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T4	2	2	-
T5	2	3	AL_Event.ind
T6	3	1	DL_EventConf.req
T7	1	1	AL_Event.ind

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
Diag_Portx	Bool	Le jeu d'Événements contient des informations de diagnostic détaillées.
Events_done	Bool	Le jeu d'Événements est traité.
Events_left	Bool	Le jeu d'Événements n'est pas encore terminé

8.3.3.2 Diagramme d'états d'Événements de la couche AL du Dispositif

La Figure 63 illustre le diagramme d'états d'Événements de la couche d'application du Dispositif.



Légende

Anglais	Français
Initialization	Initialisation
Activate/	Activation/
Deactivate/	Désactivation/

Figure 63 – Diagramme d'états d'Événements de la couche AL du Dispositif

Le Tableau 75 définit les états et transitions du diagramme d'états d'Événements de la couche d'application du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 75 – États et transitions du diagramme d'états d'Événements de la couche AL du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Event_inactive_0	Le traitement d'Événements de couche AL du Dispositif est inactif.
Event_idle_1	La couche AL du Dispositif est prête à accepter des Événements AL (informations de diagnostic) à partir des applications spécifiques à la technologie du Dispositif pour transfert à la couche DL. Pendant ce temps, les applications du Dispositif peuvent générer de nouveaux Événements.
Await_event_response_2	La couche AL du Dispositif a propagé un AL_Event avec des informations de diagnostic et attend une confirmation DL_EventTrigger de la couche DL. Pendant ce temps, la couche AL du Dispositif ne doit pas accepter un quelconque AL_Event nouveau.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	0	-
T3	1	2	Une demande AL_Event déclenche un DL_Event et le service DL_EventTrigger correspondant. Le DL_Event achemine les informations de diagnostic de la couche AL à la couche DL. Le service DL_EventTrigger établit le drapeau d'Événement dans l'échange des données cycliques (voir A.1.5)
T4	2	1	Une confirmation de DL_EventTrigger déclenche une confirmation d'AL_Event.

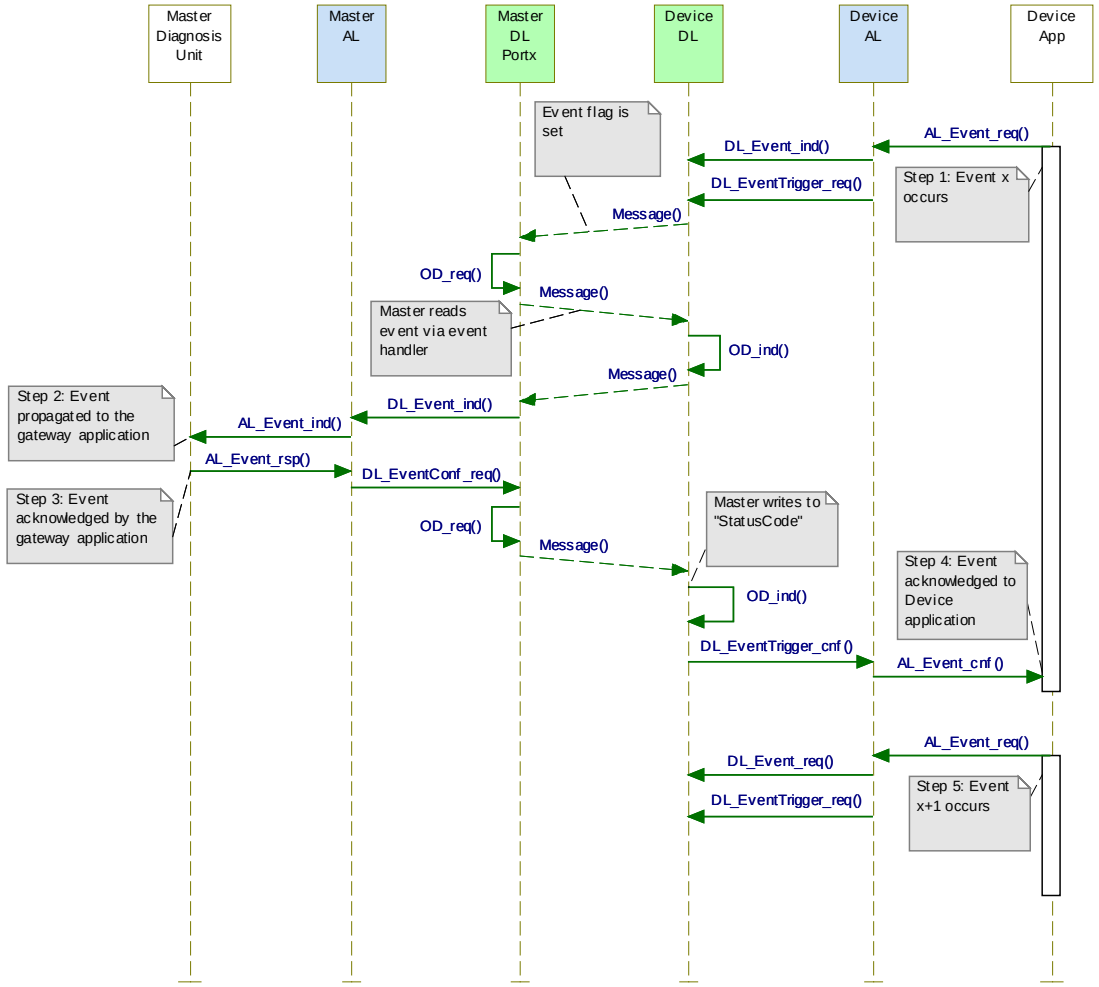
ÉLÉMENTS INTERNES	TYPE	DÉFINITION
aucun		

8.3.3.3 Planification d'Événements uniques

La Figure 64 illustre la manière dont est traité un Événement unique en provenance d'un Dispositif, conformément aux diagrammes d'états correspondants.

- L'application du Dispositif génère une demande d'Événement (Étape 1), qui est acheminée de la couche AL à la couche DL et mise en mémoire-tampon d'Événements (voir Tableau 56).
- La couche AL du Dispositif active le service EventTrigger pour établir le drapeau d'Événement et de ce fait le Maître lit l'Événement de la mémoire d'Événements.
- Le Maître propage ensuite cet Événement à l'application passerelle (Étape 2), et attend l'acquittement de l'Événement.
- Une fois l'acquittement de l'Événement reçu (Étape 3), il est indiqué au Dispositif par écriture dans le Code d'État (Étape 4).
- Le Dispositif confirme la demande d'Événement initiale à son application (Étape 5), et celle-ci peut à présent lancer une nouvelle demande d'Événement.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master Diagnosis Unit	Unité de Diagnostic du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Event flag is set	Le Drapeau d'Événement est établi
Step 1: Event x occurs	Étape 1: L'événement x a lieu
Master reads event via event handler	Le Maître lit l'événement via le module de traitement d'Événements
Step 2: Event propagated to the gateway application	Étape 2: L'événement est propagé à l'application passerelle
Step 3: Event acknowledged by the gateway application	Étape 3: L'événement est acquitté par l'application passerelle
Master writes to "Status Code"	Le Maître écrit vers "Code d'État"
Step 4: Event acknowledged to Device application	Étape 4: L'événement est acquitté par l'application du Dispositif
Step 5: Event x+1 occurs	Étape 5: L'événement x+1 a lieu

Figure 64 – Planification d'Événements uniques

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

8.3.3.4 Transport d'Événements multiples (Dispositifs patrimoniaux uniquement)

Outre la méthode décrite en 8.3.3.3 dans le cadre de laquelle chaque Événement unique est acheminé par les couches et acquitté par l'application passerelle, tous les Maîtres doivent prendre en charge ce qu'on appelle le "Transport d'Événements multiples" qui permet de transférer en une seule fois jusqu'à 6 Événements. La couche AL du Maître transfère le jeu d'Événements comme une indication de diagnostic unique à l'application passerelle et renvoie à l'application du Dispositif héritage un acquittement unique pour l'ensemble du jeu.

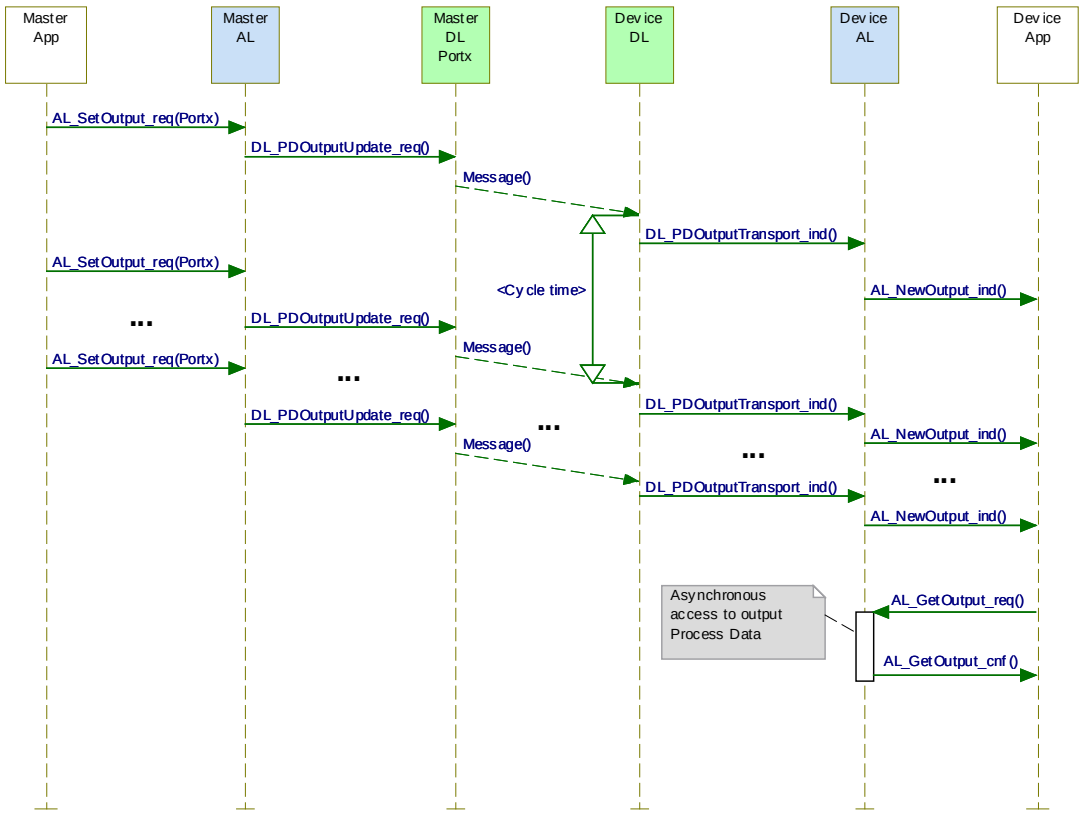
La Figure 64 s'applique également au transport d'Événements multiples, à l'exception du fait que ce transport utilise une indication DL_Event pour chaque emplacement de mémoire d'Événement et une indication unique AL_Event pour l'ensemble du jeu d'Événements.

Une demande AL_Event.req transporte jusqu'à 6 Événements et une AL_Event.ind indique jusqu'à 6 Événements en cours. Les services AL_Event.rsp et AL_Event.cnf font référence à l'ensemble du jeu d'Événements indiqué.

8.3.4 Cycles de Données de Processus

Les Figure 65 à Figure 66 montrent l'ensemble des interactions entre le Maître et le Dispositif pour plusieurs cas d'utilisation de Données de Processus de sortie et d'entrée.

La Figure 65 montre la manière dont les services AL et DL du Maître et du Dispositif sont impliqués dans l'échange cyclique de Données de Processus de sortie. L'application du Dispositif est en mesure d'acquérir les valeurs en cours des PD de sortie via le service AL_GetOutput.



Légende

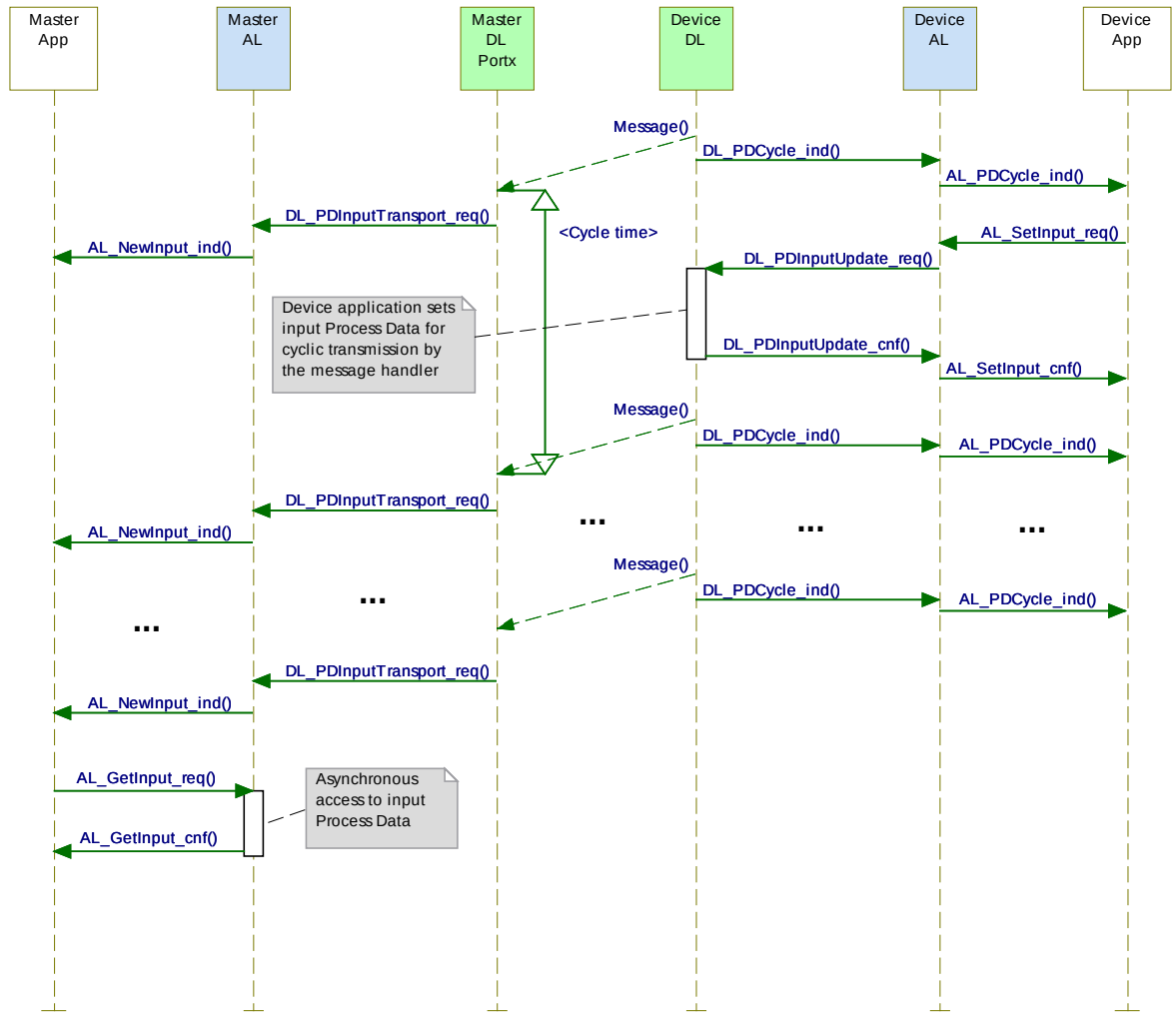
Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Asynchronous access to output Process Data	Accès asynchrone aux Données de Processus de sortie

Figure 65 – Diagramme de séquence pour des Données de Processus de sortie

La Figure 66 montre la manière dont les services AL et DL du Maître et du Dispositif sont impliqués dans l'échange cyclique de Données de Processus d'entrée. L'application du Maître est en mesure d'acquérir les valeurs en cours des PD d'entrée via le service AL_GetInput.



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Device application sets input Process Data for cyclic transmission by the message handler	L'application du Dispositif établit les Données de Processus d'entrée pour transmission cyclique par le module de traitement des messages
Asynchronous access to input Process Data	Accès asynchrone aux Données de Processus d'entrée

Figure 66 – Diagramme de séquence pour des Données de Processus d'entrée

9 Gestion système (SM)

9.1 Généralités

La Gestion Système de l'interface SDCI est chargée d'assurer le démarrage coordonné des ports du Maître, ainsi que les opérations correspondantes dans les Dispositifs connectés. La différence entre la SM du Maître et celle du Dispositif est plus significative que pour les autres couches. En conséquence, la structure de l'Article 9 traite de manière séparée les services et protocoles du Maître et du Dispositif.

9.2 Gestion Système du Maître

9.2.1 Aperçu

Les services de Gestion Système du Maître sont utilisés pour configurer les ports du Maître ainsi que le système pour tous les modes opérationnels possibles.

La SM du Maître règle les ports:

- en mettant en place la version requise des protocoles de communication;
- en vérifiant la compatibilité du Dispositif (correspondance entre les identifications réelles du Dispositif et les valeurs prévues);
- en réglant les types de séquences M appropriés du Maître et les MasterCycleTimes.

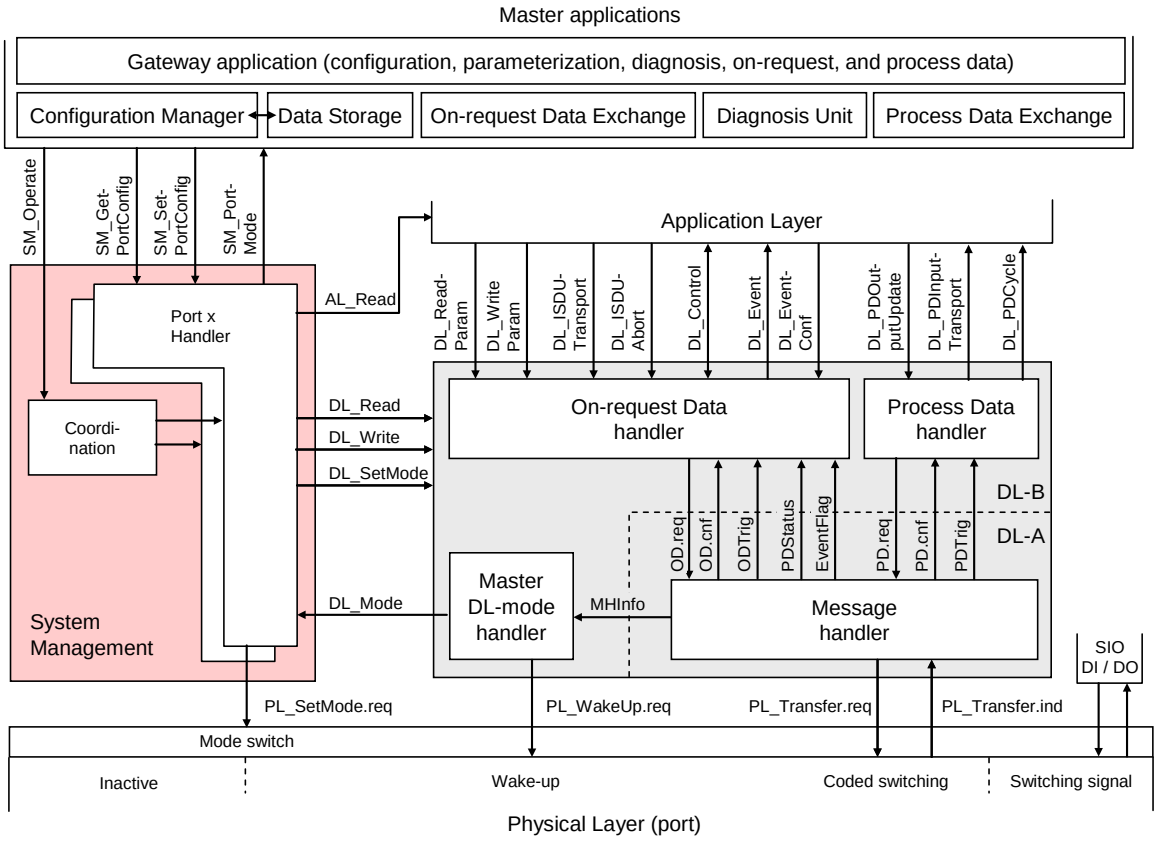
Pour ce faire, la SM utilise les services suivants illustrés en Figure 67.

- SM_SetPortConfig transfère les paramètres de Dispositif nécessaires (données de configuration) de la Gestion de la Configuration (CM) à la Gestion Système (SM). Le port démarre ensuite implicitement.
- SM_PortMode rend compte du résultat positif de la configuration du port à la CM si la configuration et la vérification du port sont correctes. Il rend compte également du résultat négatif à la CM, par l'intermédiaire d'"erreurs" correspondantes en cas de discordance des versions et d'incompatibilité des Dispositifs.
- SM_GetPortConfig lit les paramètres réels et effectifs.
- SM_Operate commute les ports en mode "OPERATE".

La Figure 67 donne un aperçu de la structure et des services de la Gestion Système du Maître.

La Gestion Système du Maître nécessite un service de couche d'application (AL_Read) pour acquérir, à des fins de vérification, les données (paramètre d'identification) à partir d'index spéciaux.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master applications	Applications du Maître
Gateway application (configuration, parameterization, diagnosis, on-request, and process data)	Application passerelle (configuration, paramétrage, diagnostic, données à la demande et de processus)
Configuration Manager	Gestionnaire de configuration
Data Storage	Stockage des données
On-request Data Exchange	Échange de données à la demande
Diagnosis Unit	Unité de Diagnostic
Process Data Exchange	Échange de Données de Processus
Application Layer	Couche d'Application
Port x Handler	Module de traitement de Port-x
On-request Data Handler	Module de traitement de Données à la Demande
Process Data Handler	Module de traitement des Données de Processus
Coordination	Coordination
System management	Gestion Système
Master DL-mode handler	Module de traitement de mode DL du Maître
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Inactive	Inactive
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer (port)	Couche Physique (port)

Figure 67 – Structure et services de la Gestion Système du Maître

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

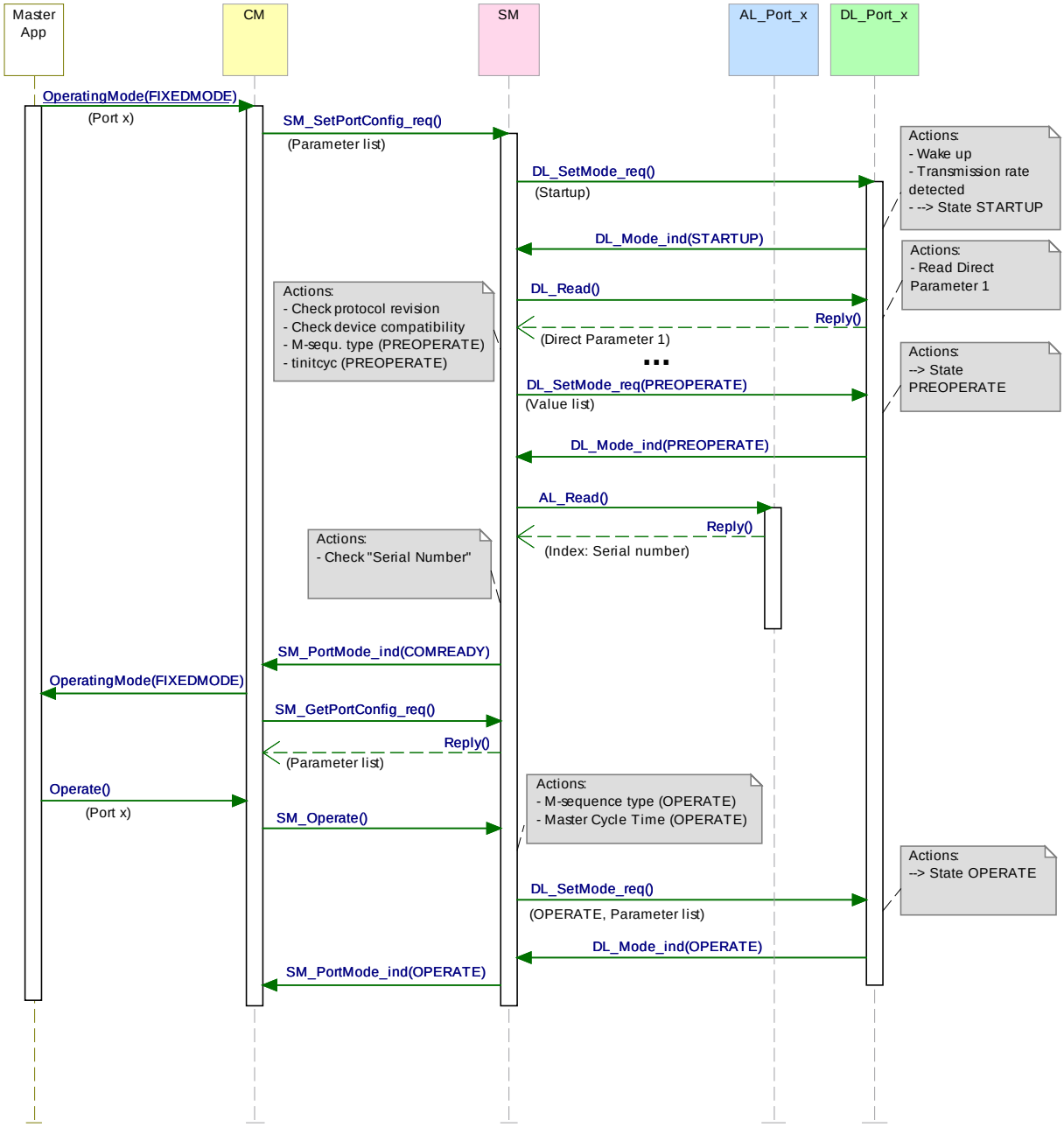
La Figure 68 illustre les actions entre les couches d'application du Maître (App Maître), de Gestion de la Configuration (CM), de Gestion Système (SM), de liaison des données (DL) et la couche d'application (AL) en cas d'utilisation en démarrage d'un port particulier.

Ce cas d'utilisation particulier est caractérisé par les postulats suivants:

- Le Dispositif pour la configuration disponible est connecté et la vérification est réussie;
- Le Dispositif utilise la version de protocole correcte conformément à sa spécification;
- Le InspectionLevel configuré est "type compatible" (Le SerialNumber est lu à partir du Dispositif et non vérifié).

Les flèches en pointillés de la Figure 68 représentent les services de réponse à un service initial.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master App	App. du Maître
(Parameter list)	(Liste de paramètres)
(Startup)	(Démarrage)
Actions:	Actions:
- Check protocol revision	- Vérification de la version du protocole
- Check device compatibility	- Vérification de la compatibilité du dispositif
- M-sequ. type (PREOPERATE)	- Type de séqu. M (PREOPERATE)
- Cycle time (PREOPERATE)	- Durée de cycle (PREOPERATE)
Reply()	Réponse()
(Direct Parameter 1)	(Page de paramètres directs 1)
- Wake up	- Réactivation

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
- Transmission rate detected	- Vitesse de transmission détectée
- --> State STARTUP	- --> État STARTUP
- Read Direct Parameter 1	- Lecture de la Page 1 de Paramètres Directs
--> State PREOPERATE	--> État PREOPERATE
(Value list)	(Liste des valeurs)
- Check "Serial Number"	- Vérification du "Numéro de série"
(Index: Serial number)	(Index: Numéro de série)
(Parameter list)	(Liste de paramètres)
- M-sequence type (OPERATE)	- Type de séquence M (OPERATE)
- Master Cycle Time (OPERATE)	- Durée de cycle du Maître (OPERATE)
--> State OPERATE	--> État OPERATE
(OPERATE, Parameter list)	(OPERATE, liste de paramètres)

Figure 68 – Organigramme séquentiel du cas d'utilisation "Configuration de port x"

9.2.2 Services SM du Maître

9.2.2.1 Aperçu

La Gestion Système fournit à l'utilisateur les services SM du Maître via son interface supérieure. Le Tableau 76 énumère les attributions du Maître quant à son rôle d'initiateur ou de destinataire des services individuels de la Gestion Système.

Tableau 76 – Services SM au sein du Maître

Désignation du service	Maître
SM_SetPortConfig	R
SM_GetPortConfig	R
SM_PortMode	I
SM_Operate	R
Légende (voir 3.3.4) I Initiateur du service R Destinataire (Répondeur) du service	

9.2.2.2 SM_SetPortConfig

Le service SM_SetPortConfig est utilisé pour définir la configuration demandée du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 77.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 77 – SM_SetPortConfig

Nom du paramètre	.req	.cnf
Argument	M	
ParameterList	M	
Résultat(+)		S
Numéro de port		M
Résultat (-)		S
Numéro de port		M
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

ParameterList
Ce paramètre contient le port configuré et les paramètres de Dispositif d'un port du Maître.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

Numéro de port
Ce paramètre contient le numéro de port

ConfiguredCycleTime
Ce paramètre contient la durée de cycle demandée pour le mode OPERATE

Valeurs autorisées:
0 (FreeRunning)
Temps (voir le Tableau B.3)

TargetMode
Ce paramètre indique le mode de fonctionnement demandé du port

Valeurs autorisées: INACTIVE, DI, DO, CFGCOM, AUTOCOM (voir le Tableau 79)

ConfiguredBaudrate:
Ce paramètre indique la vitesse de transmission demandée

Valeurs autorisées:
AUTO (Le Maître accepte la vitesse de transmission trouvée pendant "ESTABLISHCOM")
COM1 (vitesse de transmission de COM1)
COM2 (vitesse de transmission de COM2)
COM3 (vitesse de transmission de COM3)

ConfiguredRevisionID (CRID):
Longueur de données: 1 octet pour la version de protocole (voir B.1.5)

InspectionLevel:
Valeurs autorisées: NO_CHECK, TYPE_COMP, IDENTICAL (voir le Tableau 78)

ConfiguredVendorID (CVID)
Longueur de données: 2 octets

NOTE Les VendorID (identifiants de fournisseurs) sont attribués par le "IO-Link consortium"

ConfiguredDeviceID (CDID)
Longueur de données: 3 octets

ConfiguredFunctionID (CFID)
Longueur de données: 2 octets

ConfiguredSerialNumber (CSN)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Longueur de données: jusqu'à 16 octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Numéro de port
Ce paramètre contient le numéro de port

Résultat (-):
Ce paramètre de sélection indique que le service a échoué

Numéro de port
Ce paramètre contient le numéro de port

ErrorInfo
Ce paramètre contient des informations d'erreur

Valeurs autorisées:
PARAMETER_CONFLICT (cohérence du jeu de paramètres non respectée)

Le Tableau 78 spécifie le codage des différents niveaux de vérification (valeurs du paramètre InspectionLevel) (voir 9.2.3.2 et 11.8.5).

Tableau 78 – Définition de InspectionLevel (IL)

Paramètre	Niveau de vérification (IL)		
	NO_CHECK	TYPE_COMP	IDENTICAL
DeviceID (DID) (compatible)	-	Oui (RDID=CDID)	Oui (RDID=CDID)
VendorID (VID)	-	Oui (RVID=CVID)	Oui (RVID=CVID)
SerialNumber (SN)	-	-	Oui (RSN = CSN)

Le Tableau 79 spécifie le codage des différents Modes Cibles.

Tableau 79 – Définitions des Modes Cibles

Mode Cible	Définition
CFGCOM	Dispositif communiquant en mode CFGCOM après inspection réussie
AUTOCOM	Dispositif communiquant en mode AUTOCOM sans inspection
INACTIVE	Communication désactivée, pas de DI, pas de DO
DI	Port en mode d'entrée numérique (SIO)
DO	Port en mode de sortie numérique (SIO)

CFGCOM est un mode cible fondé sur une configuration utilisateur (par exemple, au moyen d'un IODD) et sur un contrôle de cohérence de RID, VID et DID.

AUTOCOM est un mode cible sans configuration. Ceci veut dire qu'il n'y a aucun contrôle de CVID et de CDID. Le CRID est établi sur la version la plus élevée prise en charge par le Maître. Il convient qu'AUTOCOM ne puisse être sélectionné qu'avec le niveau de vérification "NO_CHECK" (voir le Tableau 78).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.2.2.3 SM_GetPortConfig

Le service SM_GetPortConfig est utilisé pour acquérir la configuration courante (réelle) du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 80.

Tableau 80 – SM_GetPortConfig

Nom du paramètre	.req	.cnf
Argument	M	
Numéro de port	M	
Résultat(+)		S(=)
ParameterList		M
Résultat (-)		S(=)
Numéro de port		M
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Numéro de port
Ce paramètre contient le numéro de port

Résultat (+):
Ce paramètre de sélection indique que la demande de service a été exécutée avec succès.

ParameterList
Ce paramètre contient le port et le paramètre de Dispositif configurés d'un port du Maître.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

PortNumber
Ce paramètre contient le numéro de port.

TargetMode
Ce paramètre indique le mode de fonctionnement

Valeurs autorisées: INACTIVE, DI, DO, CFGCOM, AUTOCOM (voir le Tableau 79)

RealBaudrate
Ce paramètre indique la vitesse de transmission réelle

Valeurs autorisées:
COM1 (vitesse de transmission de COM1)
COM2 (vitesse de transmission de COM2)
COM3 (vitesse de transmission de COM3)

RealCycleTime
Ce paramètre contient la durée de cycle courant (réelle)

RealRevision (RRID)
Longueur de données: 1 octet pour la version de protocole (voir B.1.5)

RealVendorID (RVID)
Longueur de données: 2 octets

NOTE Les VendorID (identifiants de fournisseurs) sont attribués par le "IO-Link consortium"

RealDeviceID (RDID)
Longueur de données: 3 octets

RealFunctionID (RFID)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Longueur de données: 2 octets

RealSerialNumber (RSN)
Longueur de données: jusqu'à 16 octets

Résultat (-):
Ce paramètre de sélection indique que le service a échoué

Numéro de port
Ce paramètre contient le numéro de port

ErrorInfo
Ce paramètre contient des informations d'erreur

Valeurs autorisées:
PARAMETER_CONFLICT (cohérence du jeu de paramètres non respectée)

Tous les paramètres doivent être mis à "0" s'il n'y a pas d'information disponible.

9.2.2.4 SM_PortMode

Le service SM_PortMode est utilisé pour indiquer des changements ou des défauts du mode de communication local. Ceux-ci doivent être notifiés à l'application du Maître. Les paramètres des primitives de service sont énumérés dans le Tableau 81.

Tableau 81 – SM_PortMode

Nom du paramètre	.ind
Argument	M
Numéro de port	M
Mode	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Numéro de port
Ce paramètre contient le numéro de port

Mode
Valeurs autorisées:

INACTIVE	(Communication désactivée, pas de DI, pas de DO)
DI	(Port en mode d'entrée numérique (SIO))
DO	(Port en mode de sortie numérique (SIO))
COMREADY	(Communication établie et inspection réussie)
SM_OPERATE	(Le Port est prêt à échanger des Données de Processus)
COMLOST	(Échec de la communication, nouvelle procédure de réactivation requise)
REVISION_FAULT	(Version de protocole incompatible)
COMP_FAULT	(Dispositif incompatible ou Dispositif héritage selon l'InspectionLevel)
SERNUM_FAULT	(SerialNumber non concordant selon l'InspectionLevel)

9.2.2.5 SM_Operate

Le service SM_Operate sollicite la Gestion Système pour calculer les MasterCycleTimes des ports lorsqu'ils font l'objet d'un acquittement positif de Résultat (+). Ce service est en vigueur sur tous les ports. Les paramètres des primitives de service sont énumérés dans le Tableau 82.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 82 – SM_Operate

Nom du paramètre	.req	.cnf
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
TIMING_CONFLICT (la combinaison demandée de durées de cycle pour les ports activés n'est pas possible)

9.2.3 Protocole SM du Maître

9.2.3.1 Aperçu

Du fait des caractéristiques exhaustives de la configuration, du paramétrage et des fonctionnalités opérationnelles du SDCI, la description du comportement à l'aide de diagrammes d'états devient quelque peu complexe. Comme pour les diagrammes d'états de la couche DL, le 9.2.3 utilise la possibilité de disposer de sous-machines dans les diagrammes d'états principaux.

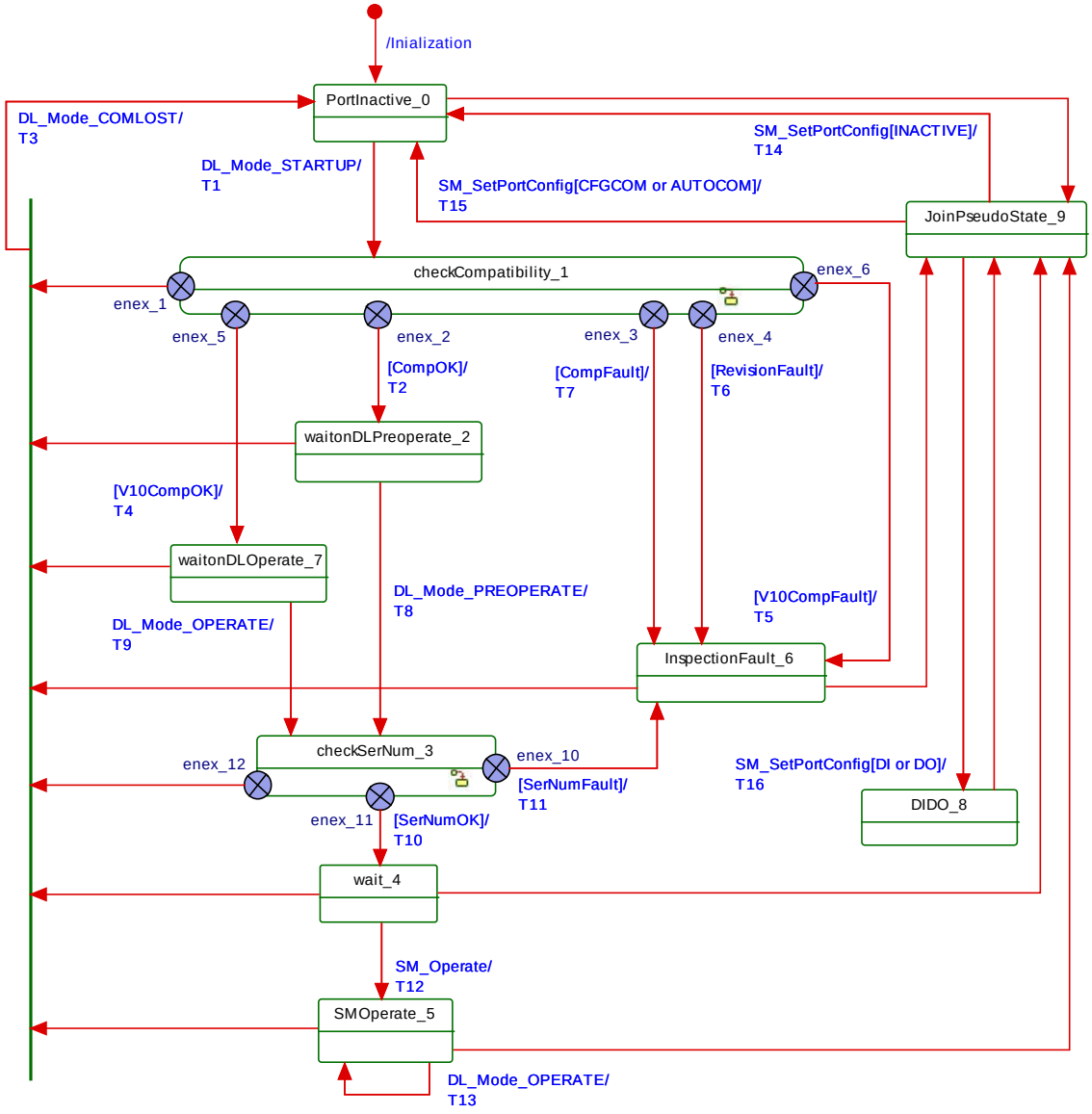
Les méthodes globales de vérification de la compatibilité sont réalisées au sein des états des sous-machines. Ces méthodes sont indiquées par les champs "*Appliquer méthode*" dans les graphes d'états, comme par exemple en Figure 70.

La logique décisionnelle correspondante est représentée par des diagrammes d'activités (voir la Figure 71, la Figure 72, la Figure 73 et la Figure 76).

9.2.3.2 Diagramme d'états de la SM du Maître

La Figure 69 illustre le diagramme d'états principal de la Gestion Système du Maître. Deux sous-machines pour la vérification de la compatibilité et du numéro de série sont décrites en 9.2.3.3 et 9.2.3.4. En cas d'interruption de la communication, la Gestion Système est informée via le service DL_Mode (COMLOST). Seul le service SM_SetPortConfig permet de reconfigurer un port. Le service SM_Operate (en vigueur sur tous les ports) n'a d'effet dans aucun état, sauf dans l'état "wait_4".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation

Figure 69 – Diagramme d'états principal de la Gestion Système du Maître

Le Tableau 83 présente les tables de transitions d'états de la Gestion Système du Maître.

Tableau 83 – Tables de transitions d'états de la Gestion Système du Maître

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
PortInactive_0	Pas de communication
CheckCompatibility_1	Le Port démarre et la version ainsi que la compatibilité du Dispositif sont vérifiées. Voir la Figure 70.
waitonDLPreoperate_2	Attendre jusqu'à ce que l'état PREOPERATE soit établi et que tous les modules de traitement des Données à la Demande aient démarré. Le port est prêt pour la communication.
CheckSerNum_3	Le numéro de série est vérifié en fonction du niveau de vérification (IL). Voir la Figure 75.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
wait_4	Le Port est prêt à communiquer et attend le service SM_Operate de la CM.
SM Operate_5	Le Port est à l'état OPERATE et effectue l'échange cyclique de données de processus.
InspectionFault_6	Le port est prêt pour la communication. Cependant, l'échange cyclique des Données de Processus ne peut être effectué à cause d'incompatibilités.
waitonDLOperate_7	Attente de l'état OPERATE demandé dans le cas où le Maître est connecté à un Dispositif héritage. Le Numéro de série peut être lu par la suite.
DIDO_8	Le Port passera en mode DI ou DO (SIO, pas de communication)
JoinPseudoState_9	Ce pseudo état est utilisé à la place d'un lien de jointure UML. Il permet l'exécution de services individuels SM_SetPortConfig en fonction de l'état du système (INACTIVE, CFGCOM, AUTOCOM, DI, ou DO)

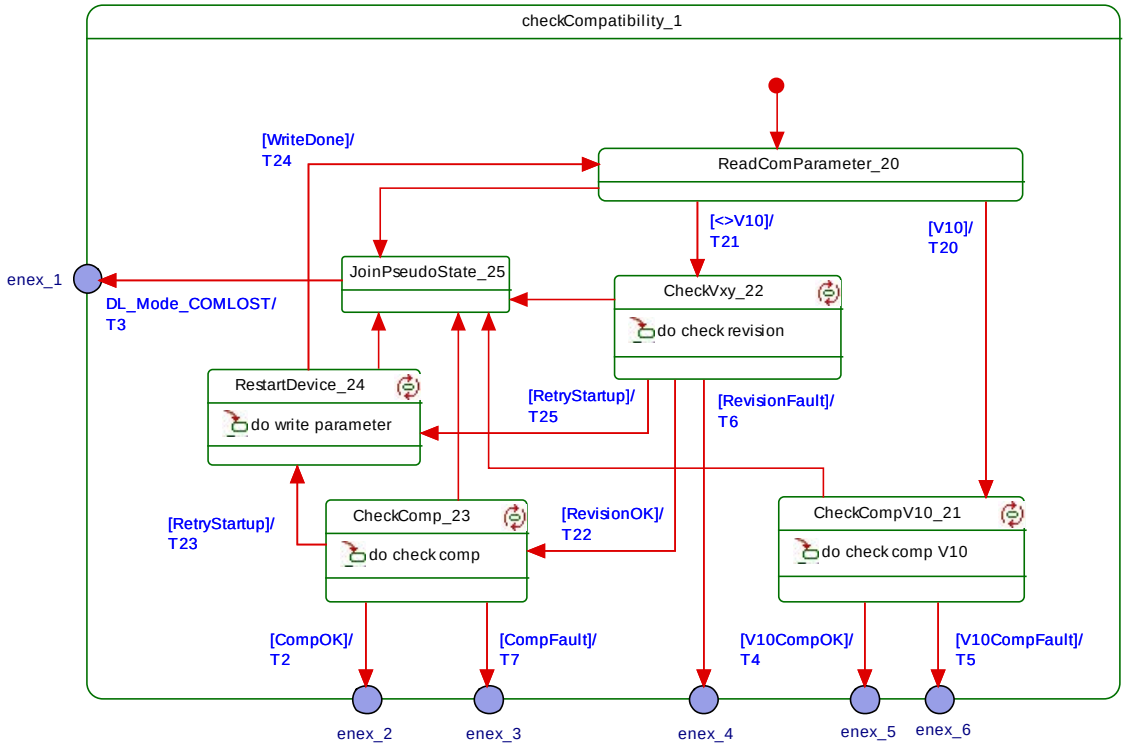
TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	CompRetry = 0
T2	1	2	DL_SetMode.req (PREOPERATE, ValueList)
T3	1,2,3,4,5, 6,7	0	DL_SetMode.req (INACTIVE) et SM_Mode.ind (COMLOST) due à un défaut de communication
T4	1	7	DL_SetMode.req (OPERATE, ValueList)
T5	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (OPERATE, ValueList)
T6	1	6	SM_PortMode.ind (REVISION_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T7	1	6	SM_PortMode.ind (COMP_FAULT), DL_SetMode.req (PREOPERATE, ValueList)
T8	2	3	-
T9	7	3	-
T10	3	4	SM_PortMode.ind (COMREADY)
T11	3	6	SM_PortMode.ind (SERNUM_FAULT)
T12	4	5	DL_SetMode.req (OPERATE, ValueList)
T13	5	5	-
T14	0,4,5,6,8	0	SM_PortMode.ind (INACTIVE), DL_SetMode.req (INACTIVE)
T15	0,4,5,6,8	0	DL_SetMode.req (STARTUP, ValueList), PL_SetMode.req (SDCI)
T16	0,4,5,6,8	8	PL_SetMode.req (SIO), SM_Mode.ind (DI ou DO), DL_SetMode.req (INACTIVE)

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
CompOK	Bool	Voir la Figure 73
CompFault	Bool	Voir la Figure 73; Variable d'erreur COMP_FAULT
RevisionFault	Bool	Voir la Figure 71; Variable d'erreur REVISION_FAULT
SerNumFault	Bool	Voir la Figure 76; Variable d'erreur SERNUM_FAULT
SerNumOK	Bool	Voir la Figure 76
V10CompFault	Bool	Voir la Figure 72; Variable d'erreur COMP_FAULT
V10CompOK	Bool	Voir la Figure 72
INACTIVE	Variable	Un Mode Cible dans le service SM_SetPortConfig
CFGCOM, AUTOCOM	Variables	Des Modes Cibles dans le service SM_SetPortConfig

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.2.3.3 Sous-machine de la SM du Maître "Vérification de la compatibilité"

La Figure 70 illustre la sous-machine de la SM du Maître checkCompatibility_1.



Légende

Anglais	Français
do check revision	Effectuer vérification de la version
do write parameter	Effectuer écriture des paramètres
do check comp	Effectuer vérification de la compatibilité
do check comp V10	Effectuer vérification de la compatibilité V10

Figure 70 – Sous-machine de la SM du Maître CheckCompatibility_1

Le Tableau 84 présente les tables de transitions d'états de la sous-machine du Maître checkCompatibility_1.

Tableau 84 – Tables de transitions d'états de la sous-machine du Maître CheckCompatibility_1

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
ReadComParameter_20	Acquisition des paramètres de communication de la page 1 de Paramètres Directs (0x02 à 0x06) via le service DL_Read (voir le Tableau B.1).
CheckCompV10_21	Acquisition des paramètres d'identification de la page 1 de Paramètres Directs (0x07 à 0x0D) via le service DL_Read (voir le Tableau B.1). Le niveau de vérification (IL) configuré définit la logique décisionnelle de la vérification de compatibilité ultérieure "CheckCompV10" avec les paramètres RVID, RDID et RFID conformément à la Figure 72.
CheckVxy_22	Une vérification est réalisée lorsque la version (CRID) configurée correspond à la version (RRID) en cours (réelle) conformément à la Figure 71.
CheckComp_23	Acquisition des paramètres d'identification de la page 1 de Paramètres Directs (0x07 à 0x0D) via le service DL_Read (voir le Tableau B.1). Le niveau de vérification (IL) configuré définit la logique décisionnelle de la vérification de compatibilité ultérieure

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
	"CheckComp" conformément à la Figure 73.
RestartDevice_24	Écriture des paramètres de compatibilité de la version de protocole configurée (CRID) et de l'identifiant de Dispositif configuré (CDID) dans le Dispositif en fonction du Mode Cible de communication CFGCOM ou AUTOCOM (voir le Tableau 79) conformément à la Figure 74.
JoinPseudoState_25	Ce pseudo état est utilisé à la place d'un lien de jointure UML. Pas de protections.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T20	20	21	-
T21	20	22	DL_Write (0x00, MCmd_MASTERIDENT), voir le Tableau B.2
T22	22	23	-
T23	23	24	-
T24	24	20	-
T25	22	24	CompRetry = CompRetry +1

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
CompOK	Bool	Voir la Figure 73
CompFault	Bool	Voir la Figure 73; Variable d'erreur COMP_FAULT
RevisionFault	Bool	Voir la Figure 71; Variable d'erreur REVISION_FAULT
RevisionOK	Bool	Voir la Figure 71
SerNumFault	Bool	Voir la Figure 76; Variable d'erreur SERNUM_FAULT
SerNumOK	Bool	Voir la Figure 76
V10	Bool	La version de protocole réelle du Dispositif connecté est une version héritée (V1.0, voir B.1.5)
<>V10	Bool	La version de protocole réelle du Dispositif connecté est conforme à la présente norme
V10CompFault	Bool	Voir la Figure 72; Variable d'erreur COMP_FAULT
V10CompOK	Bool	Voir la Figure 72
RetryStartup	Bool	Voir la Figure 71 et la Figure 73
CompRetry	Variable	Compteur interne
WriteDone	Bool	Finalisation de la séquence du service de redémarrage
MCmd_XXXXXXX	Invocation	Voir le Tableau 43

Certains états comportent des logiques complexes permettant de traiter les vérifications de compatibilité et de validité. Les Figure 71 à Figure 74 illustrent ce contexte.

La Figure 71 illustre la logique décisionnelle de la vérification de la version de protocole dans l'état "CheckVxy". Dans le cas de Dispositifs configurés, la règle suivante s'applique: si la version configurée (CRID) et la version réelle (RRID) ne correspondent pas, le CRID sera transmis au Dispositif. Si le Dispositif ne l'accepte pas, le Maître renvoie une indication via le service SM_Mode avec REV_FAULT.

Dans le cas de Dispositifs non configurés, le mode de fonctionnement AUTOCOM doit être utilisé. Pour les abréviations de noms de paramètres, voir 9.2.2.2 et 9.2.2.3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

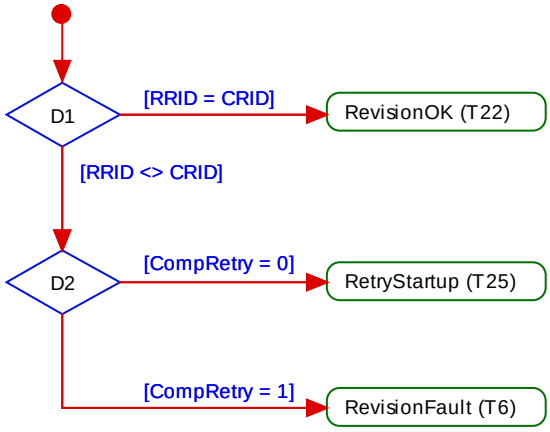
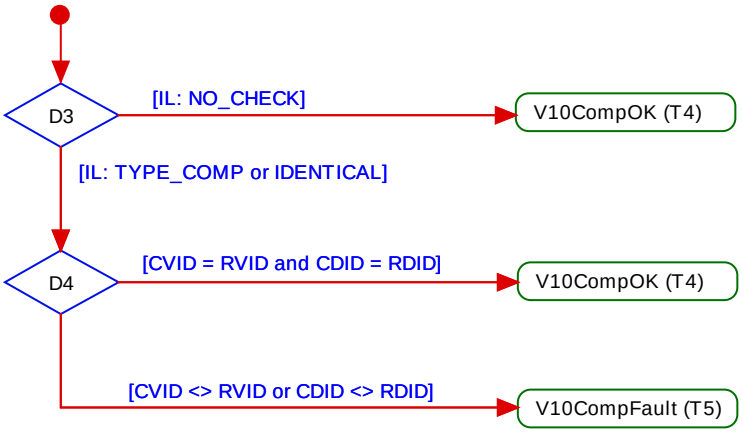


Figure 71 – Activité pour l'état "CheckVxy"

La Figure 72 illustre la logique décisionnelle pour la vérification de compatibilité de versions héritées dans l'état "CheckCompV10".



IL = Niveau de vérification

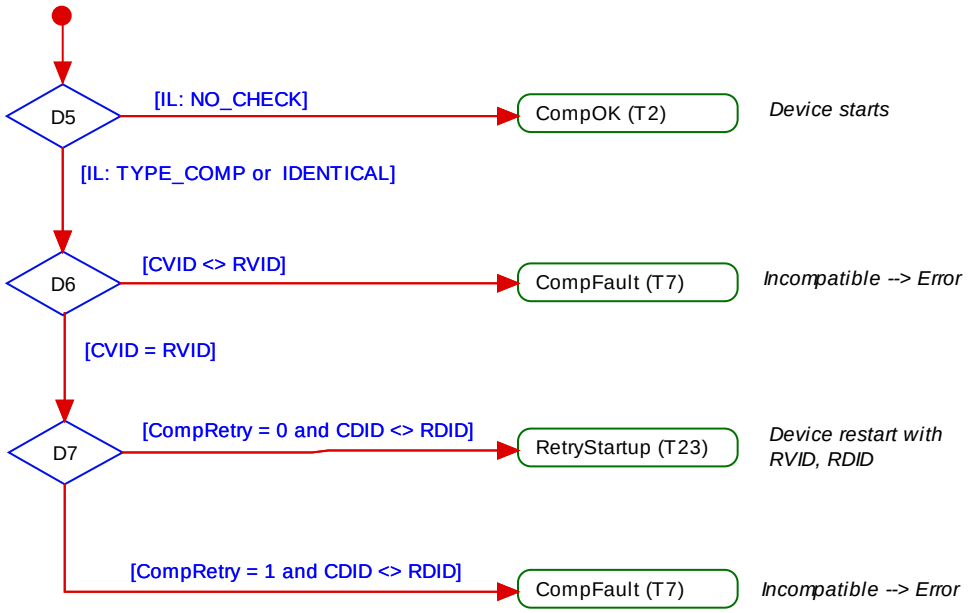
Légende

Anglais	Français
[IL: TYPE_ COMP or IDENTICAL]	[IL: TYPE_ COMP ou IDENTIQUE]
[CVID = RVID and CDID = RDID]	[CVID = RVID et CDID = RDID]

Figure 72 – Activité pour l'état "CheckCompV10"

La Figure 73 illustre la logique décisionnelle pour la vérification de compatibilité dans l'état "CheckComp".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



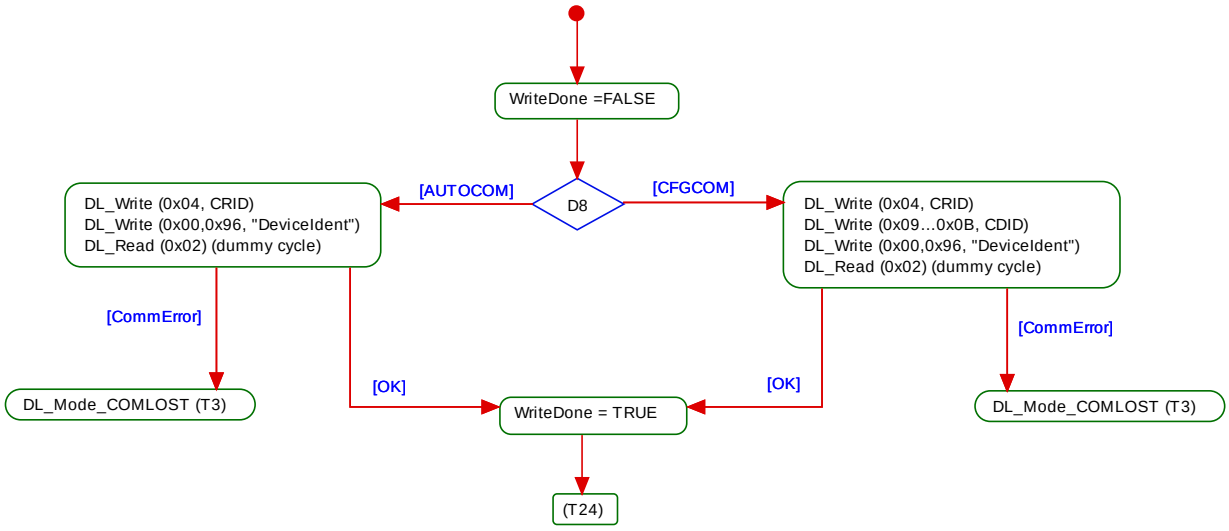
IL = Niveau de vérification

Légende

Anglais	Français
[IL: TYPE_ COMP or IDENTICAL]	[IL: TYPE_ COMP ou IDENTIQUE]
Device starts	Le Dispositif démarre
Incompatible --> Error	Incompatible --> Erreur
Device restart with RVID, RDID	Le Dispositif redémarre avec RVID, RDID

Figure 73 – Activité pour l'état "CheckComp"

La Figure 74 illustre l'activité (écriture de paramètres) dans l'état "RestartDevice".



Légende

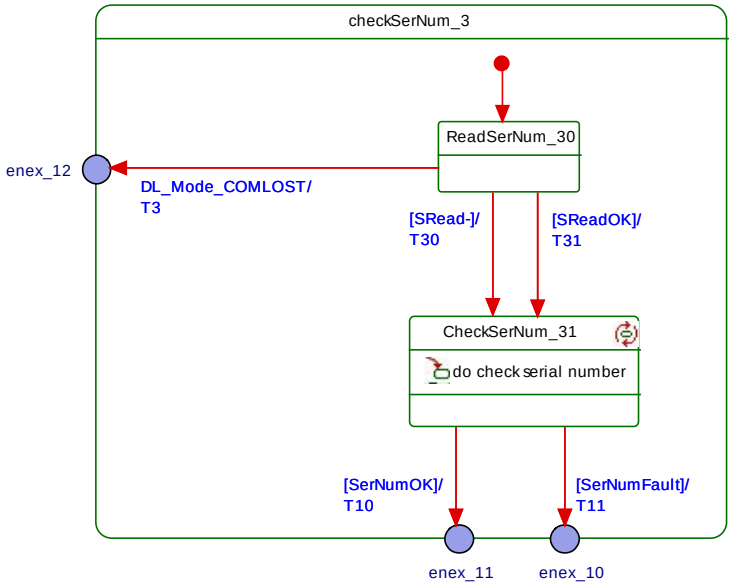
Anglais	Français
dummy cycle	cycle fictif

Figure 74 – Activité (écriture de paramètres) dans l'état "RestartDevice"

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.2.3.4 Sous-machine de la SM du Maître "Vérification du numéro de série"

La Figure 75 illustre la sous-machine de la SM du Maître "checkSerNum_3". Cette vérification est obligatoire.



Légende

Anglais	Français
do check serial number	Effectuer vérification du numéro de série

Figure 75 – Sous-machine de la SM du Maître CheckSerNum_3

Le Tableau 85 présente les tables de transitions d'états de la sous-machine du Maître CheckSerNum_3.

Tableau 85 – Table de transitions d'états de la sous-machine du Maître CheckSerNum_3

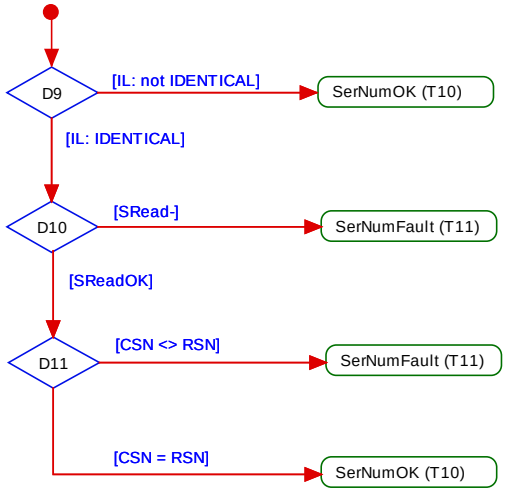
DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
ReadSerNum_30	Acquisition du Numéro de série en provenance du Dispositif via AL_Read.req (Index: 0x0015). Une réponse positive (AL_Read(+)) donne lieu à SReadOK = vrai. Une réponse négative (AL_Read(-)) donne lieu à SRead- = vrai.
CheckSerNum_31	Le Numéro de série configuré (CSN) et le Numéro de série réel (RSN) sont vérifiés en fonction du niveau de vérification (IL) conformément à la Figure 76.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T30	40	41	
T31	40	41	

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
SRead-	Bool	Réponse négative du service AL_Read (Index 0x0015)
SReadOK	Bool	Lecture de Numéro de série correcte
SerNumOK	Bool	Voir la Figure 76
SerNumFault	Bool	Voir la Figure 76

La Figure 76 illustre la logique décisionnelle (activité) pour l'état CheckSerNum_3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
[IL: not IDENTICAL]	[IL: non IDENTIQUE]
[IL: IDENTICAL]	[IL: IDENTIQUE]

Figure 76 – Activité (vérification du SerialNumber) pour l'état CheckSerNum_3

9.2.3.5 Règles d'utilisation des types de séquence M

La Gestion Système est chargée de configurer les types de séquences M corrects. Cette configuration a lieu après les actions de vérification de compatibilité (transition vers PREOPERATE) et avant le passage à OPERATE.

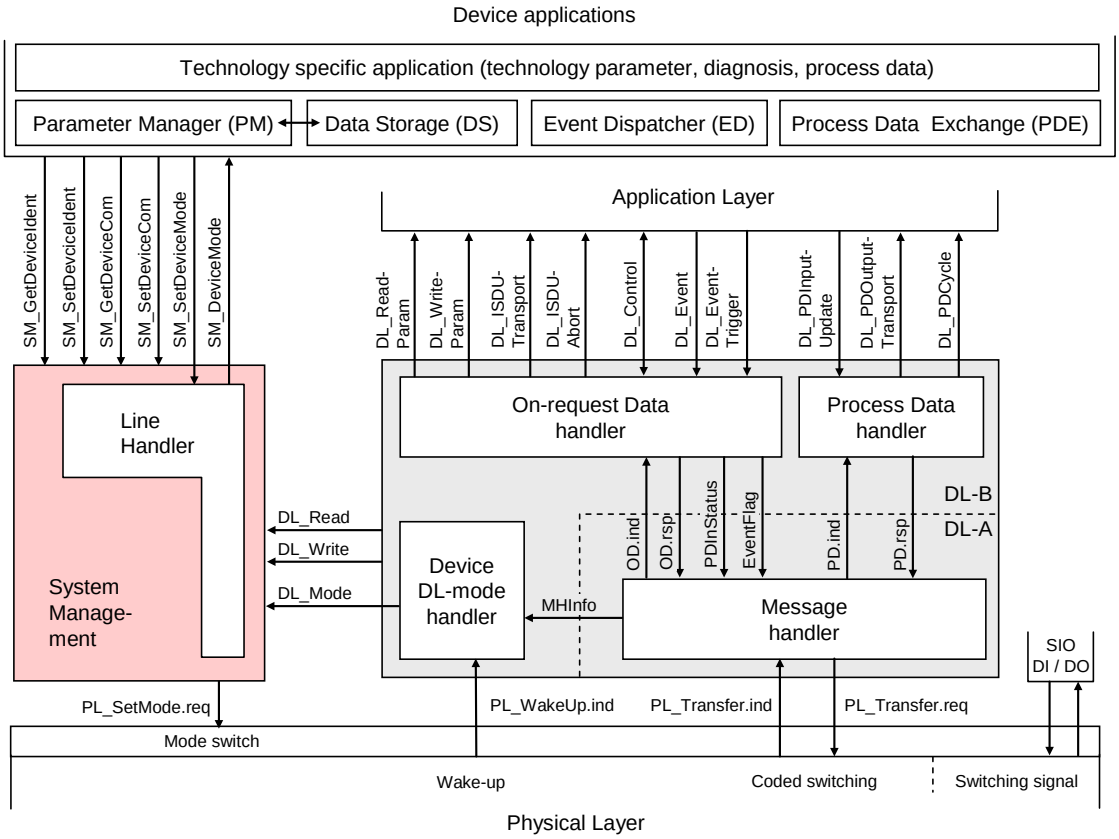
Différents types de séquences M doivent être utilisés dans les différents états opérationnels (voir A.2.6). Par exemple, lors de la commutation à l'état OPERATE, le type de séquence M pertinent pour le fonctionnement cyclique doit être utilisé. Le type de séquence M à utiliser à l'état opérationnel OPERATE est déterminé par la taille des Données de Processus d'entrée et de sortie. Les types de séquences M disponibles pour les trois modes STARTUP, PREOPERATE et OPERATE, ainsi que le codage correspondant du paramètre "Capacité de séquence M" sont spécifiés en A.2.6. Les formats de données d'entrée et de sortie doivent être obtenus à partir du Dispositif connecté afin d'ajuster le type de séquence M. Il est obligatoire qu'un Maître mette en œuvre tous les types de séquences M spécifiés en A.2.6.

9.3 Gestion Système du Dispositif

9.3.1 Aperçu

La Figure 77 illustre une vue globale de la structure et des services de la Gestion Système du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device applications	Applications du Dispositif
Technology specific application (technology parameter, diagnosis, process data)	Application spécifique à la technologie (paramètres, diagnostic, données de processus liés à la technologie)
Parameter Manager (PM)	Gestionnaire de Paramètres (PM)
Data Storage (DS)	Stockage des Données(DS)
Event Dispatcher (ED)	Répartiteur d'Événements (ED)
Process Data Exchange (PDE)	Échange de Données de Processus (PDE)
Application Layer	Couche d'Application
Line handler	Module de traitement de ligne
On-request data handler	Module de traitement de Données à la Demande
Process data handler	Module de traitement des Données de Processus
System management	Gestion Système
Device DL-mode handler	Module de traitement de mode DL du Dispositif
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer	Couche physique

Figure 77 – Structure et services de la Gestion Système (Dispositif)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Gestion Système (SM) du Dispositif fournit l'instance de contrôle centralisé par l'intermédiaire du Module de traitement de Ligne pendant toutes les phases d'initialisation, d'état par défaut (SIO), de démarrage de la communication, de communication et de repli au mode SIO.

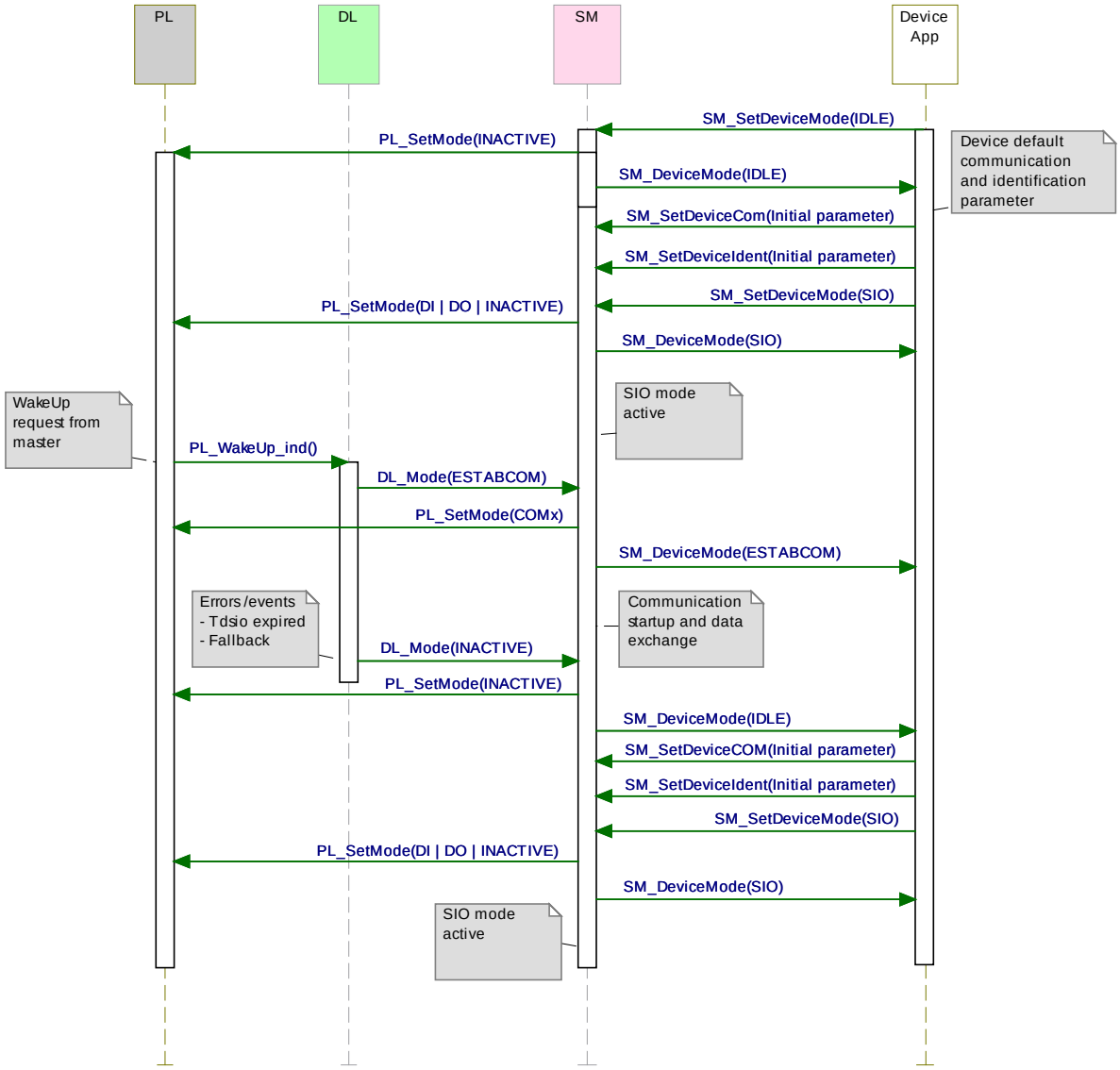
La SM du Dispositif interagit avec la couche PL afin d'établir les réglages nécessaires du circuit d'attaque de la ligne et du récepteur (voir la Figure 15), en s'appuyant sur la couche DL pour obtenir les informations nécessaires provenant du Maître (réactivation, vitesses de transmission, etc.), ainsi que sur les applications du Dispositif pour en vérifier l'identité et la compatibilité (paramètres d'identification).

Les transitions entre états du Module de traitement de Ligne (voir la Figure 79) sont initiées par les activités du port du Maître (réactivation et communication) et déclenchées par la Couche de Liaison de Données du Dispositif via les indications DL_Mode et les demandes DL_Write (commandes).

La SM fournit les paramètres d'identification du Dispositif par l'intermédiaire de l'interface d'applications du Dispositif.

L'organigramme séquentiel de la Figure 78 présente une séquence de Dispositif type, depuis l'initialisation au mode SIO par défaut, en passant par la demande de réactivation provenant du Maître jusqu'à la communication finale. L'organigramme séquentiel est complété par le cas d'utilisation d'une erreur de communication, telle que l'expiration de T_{DSIO}, un défaut de communication ou une demande de Maître comme par exemple un repli (provoqué par un Événement).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device App	Application du Dispositif
(IDLE)	(REPOS)
Device default communication and identification parameter	Paramètre de communication et d'identification par défaut du Dispositif
(Initial parameter)	(Paramètre Initial)
WakeUp request from master	Demande de réactivation en provenance du maître
SIO mode active	Mode SIO actif
Errors /events	Erreurs /Événements
- Tdsio expired	- Tdsio a expiré
- Fallback	- Repli
(INACTIVE)	(INACTIF)
Communication startup and data exchange	Démarrage de la communication et échange de données

Figure 78 – Organigramme séquentiel du cas d'utilisation "INACTIVE – SIO – SDCI – SIO"

Les services SM illustrés en Figure 78 sont définis au 9.3.2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.3.2 Services SM du Dispositif

9.3.2.1 Aperçu

Le 9.3.2 décrit les services que la Gestion Système du Dispositif fournit à ses applications, comme illustré en Figure 77.

Le Tableau 86 énumère les attributions du Dispositif quant à son rôle d'initiateur ou de destinataire du service individuel de la Gestion Système.

Tableau 86 – Services SM au sein du Dispositif

Désignation du service	Dispositif
SM_SetDeviceCom	R
SM_GetDeviceCom	R
SM_SetDeviceIdent	R
SM_GetDeviceIdent	R
SM_SetDeviceMode	R
SM_DeviceMode	I
Légende (voir 3.3.4) I Initiateur du service R Destinataire (Répondeur) du service	

9.3.2.2 SM_SetDeviceCom

Le service SM_SetDeviceCom est utilisé pour configurer les propriétés de communication prises en charge par le Dispositif dans la Gestion Système. Les paramètres des primitives de service sont énumérés dans le Tableau 87.

Tableau 87 – SM_SetDeviceCom

Nom du paramètre	.req	.cnf
Argument	M	
ParameterList	M	
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

ParameterList
Ce paramètre contient les paramètres de communication configurés pour un Dispositif donné.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

SupportedSIOMode
Ce paramètre indique le mode SIO pris en charge par le Dispositif.

Valeurs autorisées:
INACTIVE (impédance haute de la ligne C/Q),

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DI (ligne C/Q en mode d'entrée numérique),
DO (ligne C/Q en mode de sortie numérique),

SupportedTransmissionrate
Ce paramètre indique les vitesses de transmission prises en charge par le Dispositif.

Valeurs autorisées:
COM1 (vitesse de transmission de COM1)
COM2 (vitesse de transmission de COM2)
COM3 (vitesse de transmission de COM3)

MinCycleTime
Ce paramètre contient la durée de cycle minimale prise en charge par le Dispositif (voir B.1.3).

Capacité de séquence M
Ce paramètre indique les capacités prises en charge par le Dispositif (voir B.1.4):
- Prise en charge ISDU
- Types de séquences M OPERATE
- Types de séquences M PREOPERATE

RevisionID (RID)
Ce paramètre contient la version du protocole (voir B.1.5) prise en charge par le Dispositif.

ProcessDataIn
Ce paramètre contient la longueur des PD à envoyer au Maître (voir B.1.6).

ProcessDataOut
Ce paramètre contient la longueur des PD à envoyer par le Maître (voir B.1.7).

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
PARAMETER_CONFLICT (cohérence du jeu de paramètres non respectée)

9.3.2.3 SM_GetDeviceCom

Le service SM_GetDeviceCom est utilisé pour lire les propriétés de communication courantes à partir de la Gestion Système. Les paramètres des primitives de service sont énumérés dans le Tableau 88.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 88 – SM_GetDeviceCom

Nom du paramètre	.req	.cnf
Argument	M	
Résultat (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

ParameterList
Ce paramètre contient le paramètre de communication configuré pour un Dispositif.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

CurrentMode
Ce paramètre indique le mode SIO ou le mode de communication courant du Dispositif.

Valeurs autorisées:
INACTIVE (impédance haute de la ligne C/Q),
DI (ligne C/Q en mode d'entrée numérique),
DO (ligne C/Q en mode de sortie numérique),
COM1 (vitesse de transmission de COM1),
COM2 (vitesse de transmission de COM2),
COM3 (vitesse de transmission de COM3)

MasterCycleTime
Ce paramètre contient le MasterCycleTime à établir par la Gestion Système du Maître (voir B.1.3). Ce paramètre est applicable seulement à l'état SM_Operate.

Capacité de séquence M
Ce paramètre indique les capacités de la séquence M courante configurée dans la Gestion Système du Dispositif (voir B.1.4).
- Prise en charge ISDU

- Types de séquences M OPERATE

- Types de séquences M PREOPERATE

RevisionID (RID)
Ce paramètre contient la version courante du protocole (voir B.1.5) dans la Gestion Système du Dispositif.

ProcessDataIn
Ce paramètre contient la longueur courante des PD à envoyer au Maître (voir B.1.6).

ProcessDataOut
Ce paramètre contient la longueur courante des PD à envoyer par le Maître (voir B.1.7).

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (Service non disponible en l'état courant)

9.3.2.4 SM_SetDeviceIdent

Le service SM_SetDeviceIdent est utilisé pour configurer les données d'identification du Dispositif dans la Gestion Système. Les paramètres des primitives de service sont énumérés dans le Tableau 89.

Tableau 89 – SM_SetDeviceIdent

Nom du paramètre	.req	.cnf
Argument	M	
ParameterList	M	
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

ParameterList
Ce paramètre contient le paramètre d'identification configuré pour un Dispositif.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

VendorID (VID)
Ce paramètre contient le VendorID attribué à un Dispositif (voir B.1.8)

Longueur de données: 2 octets

DeviceID (DID)
Ce paramètre contient un des DeviceID attribués (voir B.1.9)

Longueur de données: 3 octets

FunctionID (FID)
Ce paramètre contient un des FunctionID attribués (voir B.1.10)

Longueur de données: 2 octets

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (service non disponible en l'état courant),
PARAMETER_CONFLICT (cohérence du jeu de paramètres non respectée)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

9.3.2.5 SM_GetDeviceIdent

Le service SM_GetDeviceIdent est utilisé pour lire le paramètre d'identification du Dispositif à partir de la Gestion Système. Les paramètres des primitives de service sont énumérés dans le Tableau 90.

Tableau 90 – SM_GetDeviceIdent

Nom du paramètre	.req	.cnf
Argument	M	
Résultat (+)		S
ParameterList		M
Result (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

ParameterList
Ce paramètre contient les paramètres de communication configurés du Dispositif.

Type de paramètre: Enregistrement

Enregistrement d'éléments:

VendorID (VID)
Ce paramètre contient le VendorID réel du Dispositif (voir B.1.8)

Longueur de données: 2 octets

DeviceID (DID)
Ce paramètre contient le DeviceID réel du Dispositif (voir B.1.9)

Longueur de données: 3 octets

FunctionID (FID)
Ce paramètre contient le FunctionID réel du Dispositif (voir B.1.10)

Longueur de données: 2 octets

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (service non disponible en l'état courant)

9.3.2.6 SM_SetDeviceMode

Le service SM_SetDeviceMode est utilisé pour mettre le Dispositif dans un état opérationnel défini lors de l'initialisation. Les paramètres des primitives de service sont énumérés dans le Tableau 91.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 91 – SM_SetDeviceMode

Nom du paramètre	.req	.cnf
Argument	M	
Mode	M	
Résultat(+)		S
Résultat (-)		S
ErrorInfo		M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Mode
Valeurs autorisées:
IDLE (Le Dispositif passe en attente de configuration)
SIO (Le Dispositif passe au mode défini dans le service "SM_SetDeviceCom")

Résultat (+):
Ce paramètre de sélection indique que le service a été exécuté avec succès.

Résultat (-):
Ce paramètre de sélection indique que le service a échoué.

ErrorInfo
Ce paramètre contient des informations d'erreur.

Valeurs autorisées:
STATE_CONFLICT (service non disponible en l'état courant)

9.3.2.7 SM_DeviceMode

Le service SM_DeviceMode est utilisé pour indiquer des changements d'états de communication à l'application du Dispositif. Les paramètres des primitives de service sont énumérés dans le Tableau 92.

Tableau 92 – SM_DeviceMode

Nom du paramètre	.ind
Argument	M
Mode	M

Argument
Les paramètres spécifiques au service sont transmis dans l'argument.

Mode
Valeurs autorisées:
IDLE (Le Dispositif passe en attente de configuration)
SIO (Le Dispositif passe au mode défini dans le service "SM_SetDeviceCom")
ESTABCOM (Le Dispositif passe en mode SM "SM_ComEstablish")
COM1 (Le Dispositif passe en mode COM1)
COM2 (Le Dispositif passe en mode COM2)
COM3 (Le Dispositif passe en mode COM3)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

STARTUP	(Le Dispositif passe en mode STARTUP)
IDENT_STARTUP	(Le Dispositif passe en mode SM "SM_IdentStartup")
IDENT_CHANGE	(Le Dispositif passe en mode SM "SM_IdentCheck")
PREOPERATE	(Le Dispositif passe en mode PREOPERATE)
OPERATE	(Le Dispositif passe en mode OPERATE)

9.3.3 Protocole SM du Dispositif

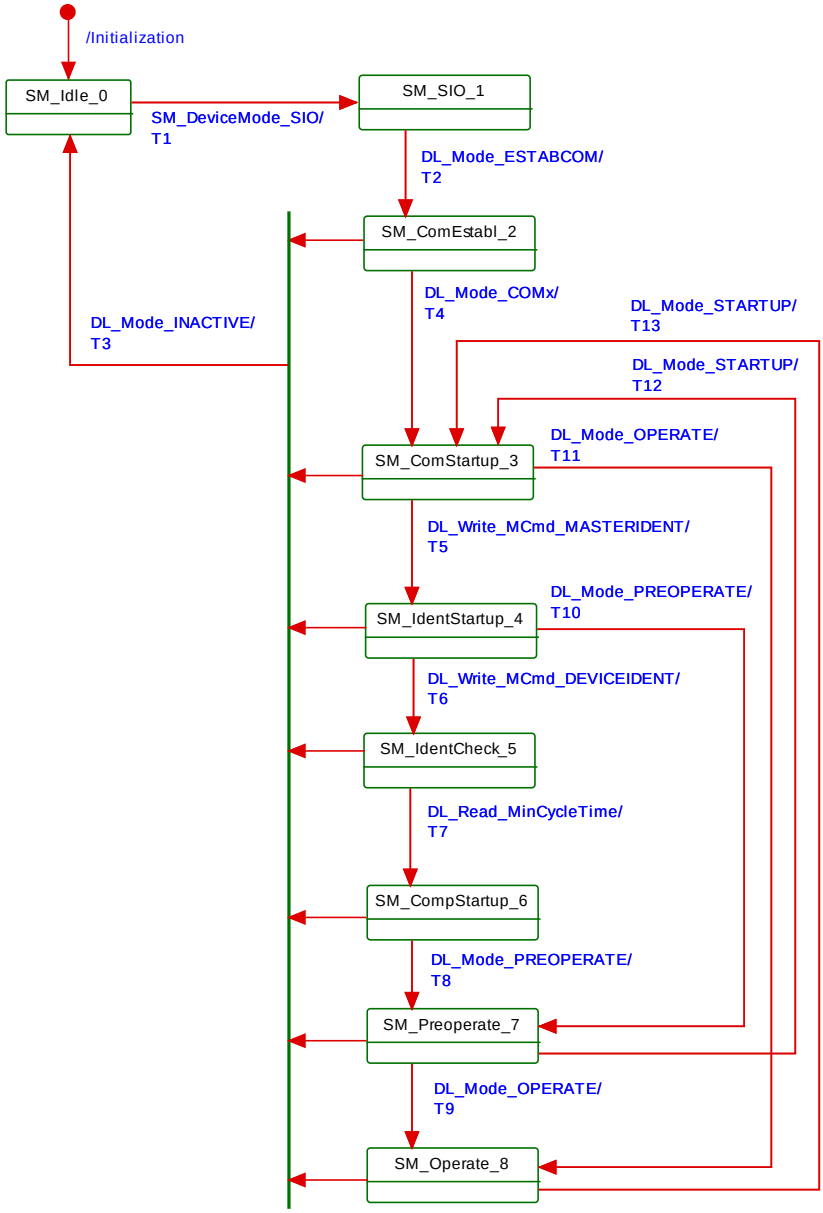
9.3.3.1 Aperçu

Le comportement du Dispositif est principalement piloté par les messages du Maître.

9.3.3.2 Diagramme d'états de la SM du Dispositif

La Figure 79 illustre le diagramme d'états du module de traitement de ligne SM du Dispositif. Il est déclenché par le module de traitement de Mode DL et l'application du Dispositif. Il évalue les différentes phases de communication au cours du démarrage et contrôle l'état de la ligne du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation

Figure 79 – Diagramme d'états de la Gestion Système du Dispositif

Le Tableau 93 définit les états individuels et les actions dans les transitions.

Tableau 93 – Table de transitions d'états de la Gestion Système du Dispositif

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
SM_Idle_0	En l'état SM_Idle, la SM est en attente de configuration par l'application du Dispositif et d'établissement du mode SIO. Cet état est abandonné à la réception d'une demande SM_SetDeviceMode(SIO) provenant de l'application du Dispositif La séquence suivante de services doit être exécutée entre l'application du Dispositif et la SM. Invocation de SM_SetDeviceCom(Liste de paramètres initiale) Invocation de SM_SetDeviceIdent(VID, DID initial, FID)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
SM_SIO_1	En l'état SM_SIO, le module de traitement de ligne SM reste en mode SIO par défaut. La Couche Physique est réglée sur les caractéristiques du mode SIO définies par l'application du Dispositif via le service SetDeviceMode. Cet état est abandonné à la réception d'une indication DL_Mode(ESTABCOM).
SM_ComEstablish_2	En l'état SM_ComEstablish, la SM est en attente d'établissement de la communication dans la Couche de Liaison de Données. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE) ou DL_Mode(COMx), où COMx peut être COM1, COM2 ou COM3.
SM_ComStartup_3	En l'état SM_ComStartup, les paramètres de communication (page 1 de Paramètres Directs, adresses 0x02 à 0x06) sont lus par la SM du Maître via des demandes DL_Read. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE), d'une indication DL_Mode(OPERATE) (Maître héritage uniquement), ou d'une demande DL_Write(MCmd_MASTERIDENT) (Maître conforme à la présente norme).
SM_IdentStartup_4	En l'état SM_IdentStartup, les données d'identification (VID, DID, FID) sont lues et vérifiées par le Maître. En cas d'incompatibilités, la SM du Maître écrit la version SDCI (RID) prise en charge et le DeviceID (DID) configuré vers le Dispositif. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE), d'une indication DL_Mode(PREOPERATE) (vérification de compatibilité réussie), ou une demande DL_Write(MCmd_DEVICEIDENT) (nouvelle compatibilité demandée).
SM_IdentCheck_5	En l'état SM_IdentCheck, la SM attend une nouvelle initialisation des paramètres de communication et d'identification. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE) ou une demande DL_Read(page 1 de Paramètres Directs, adresses 0x02 = "MinCycleTime"). Dans cet état, l'application du Dispositif doit vérifier les paramètres RID et DID à partir de la SM et régler ces données aux valeurs prises en charge. En conséquence, la séquence suivante de services doit être exécutée entre l'application du Dispositif et la SM. Invocation de SM_GetDeviceCom(RID configuré, liste de paramètres) Invocation de SM_GetDeviceIdent(DID configuré, liste de paramètres) Invocation de l'application du Dispositif qui vérifie et fournit la fonction de compatibilité et les paramètres Invocation de SM_SetDeviceCom(nouveau RID pris en charge, nouvelle liste de paramètres) Invocation de SM_SetDeviceIdent(nouveau DID pris en charge, liste de paramètres)
SM_CompStartup_6	En l'état SM_CompatStartup, les données de communication et d'identification sont relues et vérifiées par la SM du Maître. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE) ou DL_Mode(PREOPERATE).
SM_Preoperate_7	Au cours de l'état SM_Preoperate, le Numéro de série peut être lu et vérifié par la SM du Maître; de même que le Stockage des données et le paramétrage du Dispositif peuvent être exécutés. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE), DL_Mode(STARTUP) ou DL_Mode(OPERATE).
SM_Operate_8	Au cours de l'état SM_Operate, l'échange cyclique de Données de Processus et le transfert acyclique de Données à la Demande sont actifs. Cet état est abandonné à la réception d'une indication DL_Mode(INACTIVE) ou DL_Mode(STARTUP).

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	Le Dispositif passe en mode SIO configuré à la réception de la demande de déclenchement SM_SetDeviceMode.req(SIO). Invocation de PL_SetMode(DI|DO|INACTIVE) Invocation de SM_DeviceMode(SIO)
T2	1	2	Le Dispositif passe en mode de communication à la réception de l'indication de déclenchement DL_Mode.ind(ESTABCOM). Invocation de PL_SetMode(COMx) Invocation de SM_DeviceMode(ESTABCOM)
T3	2,3,4,5,6, 7,8	0	Le Dispositif passe en mode SM_Idle mode à la réception de l'indication de déclenchement DL_Mode.ind(INACTIVE). Invocation de PL_SetMode(INACTIVE) Invocation de SM_DeviceMode(IDLE)
T4	2	3	L'application du Dispositif reçoit une indication concernant le débit en bauds d'établissement de la communication dans la couche DL, déclenché par DL_Mode.ind(COMx). Invocation de SM_DeviceMode(COMx)

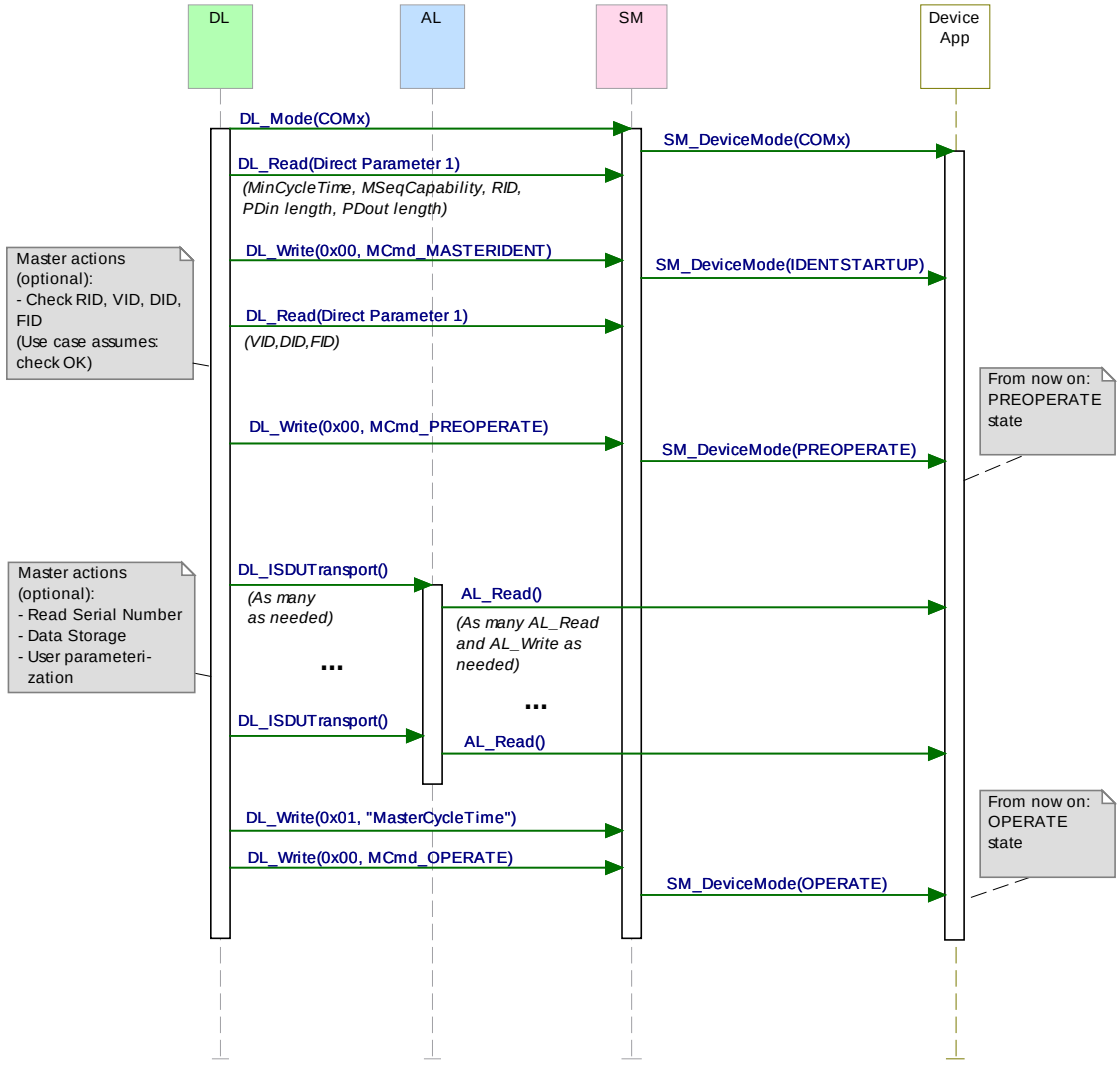
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T5	3	4	La phase d'identification du Dispositif commence à la réception de l'indication de déclenchement DL_Write.ind(MCmd_MASTERIDENT). Invocation de SM_DeviceMode(IDENTSTARTUP)
T6	4	5	La phase de vérification de l'identité du Dispositif commence à la réception de l'indication de déclenchement DL_Write.ind(MCmd_DEVICEIDENT). Invocation de SM_DeviceMode(IDENTCHANGE)
T7	5	6	La phase de départ de la vérification de la compatibilité du Dispositif commence à la réception de l'indication de déclenchement DL_Read.ind (page 1 de Paramètres Directs, adresse 0x02 = "MinCycleTime").
T8	6	7	Le Dispositif entre dans la phase de pré-fonctionnement lorsqu'il reçoit le déclencheur DL_Mode.ind(PREOPERATE). Invocation de SM_DeviceMode(PREOPERATE)
T9	7	8	Le Dispositif entre dans la phase de fonctionnement lorsqu'il reçoit le déclencheur DL_Mode.ind(OPERATE). Invocation de SM_DeviceMode(OPERATE)
T10	4	7	Le Dispositif entre dans la phase de pré-fonctionnement lorsqu'il reçoit le déclencheur DL_Mode.ind(PREOPERATE). Invocation de SM_DeviceMode(PREOPERATE)
T11	3	8	Le Dispositif entre dans la phase de fonctionnement lorsqu'il reçoit le déclencheur DL_Mode.ind(OPERATE). Invocation de SM_DeviceMode(OPERATE)
T12	7	3	Le Dispositif entre dans la phase de démarrage de la communication lorsqu'il reçoit le déclencheur DL_Mode.ind(STARTUP). Invocation de SM_DeviceMode(STARTUP)
T13	8	3	Le Dispositif entre dans la phase de démarrage de la communication lorsqu'il reçoit le déclencheur DL_Mode.ind(STARTUP). Invocation de SM_DeviceMode(STARTUP)

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
COMx	Variable	N'importe quelle vitesse de transmission: COM1, COM2 ou COM3
DL_Write_MCmd_xxx	Service	Le service DL écrit des MasterCommands (xxx = valeurs tirées du Tableau B.2)

La Figure 80 illustre un organigramme séquentiel type pour le démarrage de la communication de la SM d'un Dispositif dont les réglages correspondent à la configuration du port du Maître (démarrage ordinaire).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

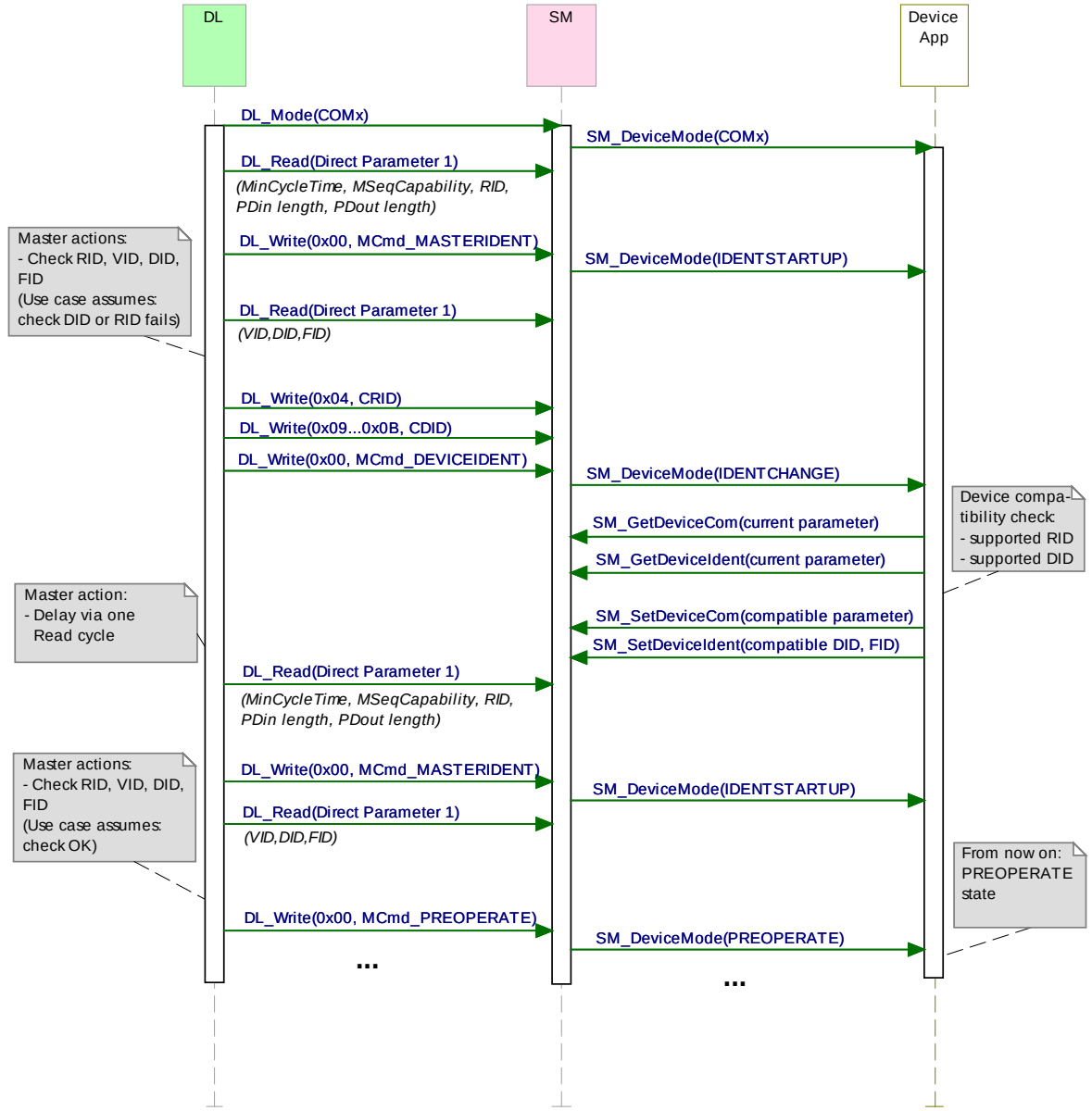
Anglais	Français
Device App	Application du Dispositif
(Direct Parameter 1)	(Page de paramètres directs 1)
(MinCycleTime, MSeqCapability, RID, PDin length, PDout length)	(MinCycleTime, MSeqCapability, RID, longueur de PDin, longueur de PDout)
Master actions	Actions du Maître
(optional):	(facultatif):
- Check RID, VID, DID,FID	- Vérification de RID, VID, DID,FID
(Use case assumes: check OK)	(Hypothèse du cas d'utilisation: Vérification OK)
From now on: PREOPERATE state	À partir de là: État PREOPERATE
Master actions (optional):	Actions du Maître (facultatif):
- Read Serial Number	- Lecture du Numéro de série
- Data Storage	- Stockage des données
- User parameterization	- Paramétrage de l'utilisateur
(As many as needed)	(Autant que nécessaire)
(As many AL_Read and AL_Write as needed)	(Autant de AL_Read et AL_Write que nécessaire)
From now on: OPERATE state	À partir de là: État OPERATE

Figure 80 – Organigramme séquentiel de démarrage ordinaire du Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Figure 81 illustre un organigramme séquentiel type pour le démarrage de la communication de la SM d'un Dispositif dont les réglages ne correspondent pas à la configuration du port du Maître (mode compatibilité). Dans ce mode, le Maître tente d'écraser les paramètres d'identification du Dispositif pour parvenir à un mode compatible et réalisable.

L'organigramme séquentiel de la Figure 81 illustre uniquement les actions jusqu'à l'état PREOPERATE. Les actions restantes jusqu'à l'état OPERATE peuvent être déduites de la Figure 80.



Légende

Anglais	Français
Device App	Application du Dispositif
(Direct Parameter 1)	(Page de paramètres directs 1)
(MinCycleTime, MSeqCapability, RID, PDin length, PDout length)	(MinCycleTime, MSeqCapability, RID, longueur de PDin, longueur de PDout)
Master actions:	Actions du Maître:
- Check RID, VID, DID,FID	- Vérification de RID, VID, DID,FID

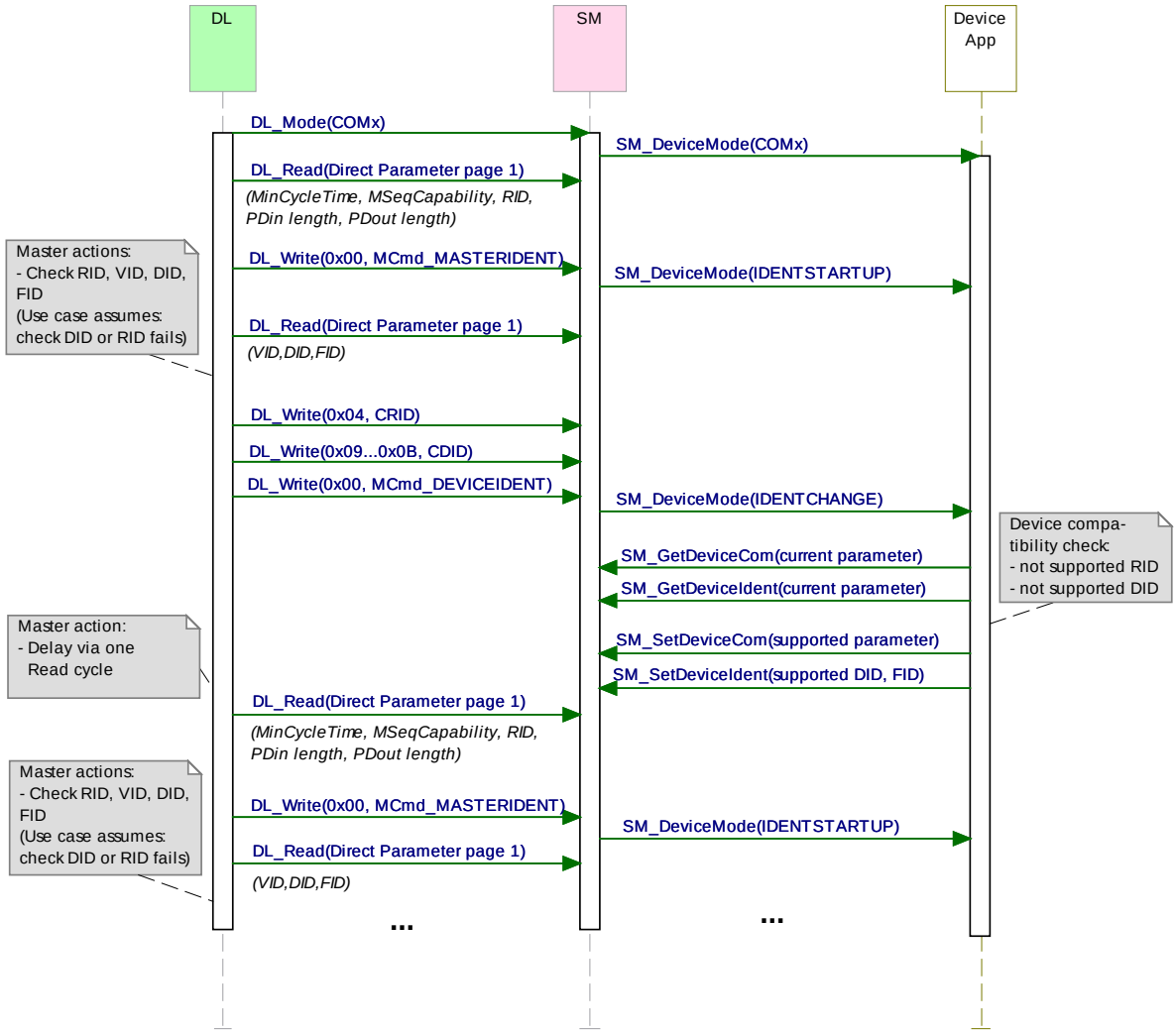
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
(Use case assumes: check DID or RID fails)	(Hypothèse du cas d'utilisation: Échec de la vérification DID ou RID)
(current parameter)	(paramètre courant)
(compatible parameter)	(paramètre compatible)
(compatible DID, FID)	(DID, FID compatible)
Master action:	Action du Maître:
- Delay via one Read cycle	- Retarder via un cycle de lecture
Device compatibility check:	- Vérification de la compatibilité du dispositif:
- supported RID	- RID pris en charge
- supported DID	- DID pris en charge
Master actions:	Actions du Maître:
- Check RID, VID, DID, FID	- Vérification de RID, VID, DID,FID
(Use case assumes: check OK)	(Hypothèse du cas d'utilisation: Vérification OK)
From now on: OPERATE state	À partir de là: État OPERATE

Figure 81 – Organigramme séquentiel du démarrage d'un Dispositif en mode compatibilité

La Figure 82 illustre un organigramme séquentiel type pour le démarrage de la communication de la SM d'un Dispositif dont les réglages ne correspondent pas à la configuration du port du Maître. La Gestion Système du Maître tente de reconfigurer le Dispositif avec d'autres paramètres d'identification du Dispositif (mode compatibilité). Dans le présent cas d'utilisation, les paramètres de remplacement sont supposés être incompatibles.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Device App	Application du Dispositif
(Direct Parameter 1)	(Page de paramètres directs 1)
(MinCycleTime, MSeqCapability, RID, PDin length, PDout length)	(MinCycleTime, MSeqCapability, RID, longueur de PDin, longueur de PDout)
Master actions:	Actions du Maître:
- Check RID, VID, DID, FID	- Vérification de RID, VID, DID, FID
(Use case assumes: check DID or RID fails)	(Hypothèse du cas d'utilisation: Échec de la vérification DID ou RID)
(current parameter)	(paramètre courant)
(supported parameter)	(paramètre pris en charge)
(supported DID, FID)	(DID, FID pris en charge)
Device compatibility check:	- Vérification de la compatibilité du dispositif:
- not supported RID	- RID non pris en charge
- not supported DID	- DID non pris en charge
Master action:	Action du Maître:
- Delay via one Read cycle	- Retarder via un cycle de lecture

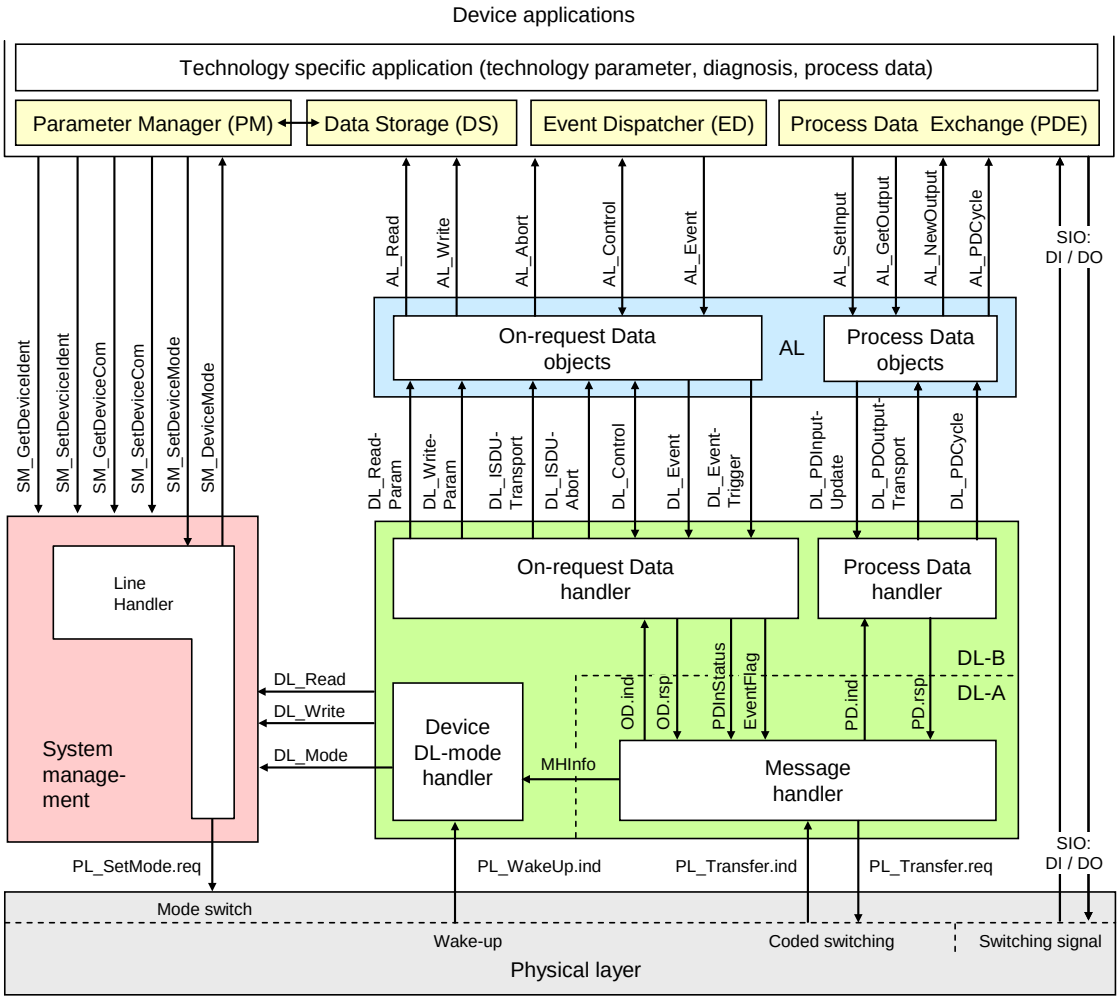
Figure 82 – Organigramme séquentiel du démarrage d'un Dispositif en cas d'échec de la compatibilité

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10 Dispositif

10.1 Aperçu

La Figure 83 illustre une vue globale de l'ensemble de la structure et des services d'un Dispositif.



Légende

Anglais	Français
Device applications	Applications du Dispositif
Technology specific application (technology parameter, diagnosis, process data)	Application spécifique à la technologie (paramètres, diagnostic, données de processus liés à la technologie)
Parameter Manager (PM)	Gestionnaire de Paramètres (PM)
Data Storage (DS)	Stockage des Données (DS)
Event Dispatcher (ED)	Répartiteur d'Événements (ED)
Process Data Exchange (PDE)	Échange de Données de Processus (PDE)
On-request Data objects	Objets de Données à la Demande
Process Data objects	Objets de Données de Processus
Line handler	Module de traitement de ligne
On-request data handler	Module de traitement de Données à la Demande
Process data handler	Module de traitement des Données de Processus

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
System management	Gestion Système
Device DL-mode handler	Module de traitement de mode DL du Dispositif
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer	Couche physique

Figure 83 – Structure et services d'un Dispositif

Les applications d'un Dispositif comprennent, en premier lieu, l'application spécifique à la technologie qui est constituée du transducteur ainsi que de ses paramètres technologiques, ses informations de diagnostic et ses Données de Processus. Les applications communes d'un Dispositif comprennent

- un Gestionnaire de Paramètres (PM) chargé des vérifications de la compatibilité et du bien-fondé de l'ensemble des jeux de paramètres spécifiques à la technologie (fournisseur) et communs au système (voir 10.3),
- un mécanisme de Stockage des Données(DS) qui peut optionnellement télécharger des paramètres à partir du Maître ou vers le Maître (voir 10.4),
- un Répartiteur d'Événements (ED) qui supervise les états et achemine des informations de diagnostic telles que les notifications, les avertissements, les erreurs ainsi que les demandes du Dispositif telles les initiatives périphériques (voir 10.5);
- une unité d'Échange de Données de Processus (PDE) qui conditionne les structures de données pour la transmission dans le cas d'un capteur ou qui prépare les structures de données reçues pour la génération de signaux. Cette unité contrôle également les états opérationnels afin de garantir la validité des Données de Processus (voir 10.2).

Ces applications de Dispositif fournissent des méthodes/fonctions et des paramètres normalisés communs à tous les Dispositifs, ainsi que des fonctions et des paramètres spécifiques au Dispositif, tous ces éléments étant définis dans l'Article 10.

10.2 Échange de Données de Processus (PDE)

L'unité d'Échange de Données de Processus transmet et reçoit de manière cyclique des Données de Processus sans interférence avec les Données à la Demande (paramètres, commandes et Événements).

Un actionneur (Données de Processus de sortie) doit observer l'émission cyclique et passer à un état par défaut approprié, pour garder par exemple la dernière valeur, s'arrêter ou se désactiver dès que l'émission de données est interrompue (voir 7.3.3.5 et 10.7.3). L'actionneur doit attendre la MasterCommand "ProcessDataOutputOperate" (voir le Tableau B.2, Données de Processus de sortie "valides") avant tout fonctionnement normal après redémarrage en cas d'interruption.

Au cours de l'échange des données cycliques, un actionneur (Données de Processus de sortie) reçoit une Commande du Maître "DeviceOperate" lorsque les Données de Processus de sortie ne sont pas valides et une Commande du Maître "ProcessDataOutputOperate" lorsqu'elles sont de nouveau valides (voir le Tableau B.2).

Il n'est pas nécessaire qu'un Dispositif capteur (Données de Processus d'entrée) surveille l'échange des données cycliques. Cependant, si le Dispositif est incapable de garantir des Données de Processus valides, l'état des PD "Données de Processus non valides" (voir A.1.5) doit être signalé à l'application du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.3 Gestionnaire de Paramètres (PM)

10.3.1 Généralités

Un Dispositif peut être paramétré selon deux méthodes de base, en utilisant les Paramètres Directs ou l'espace mémoire des index accessible au moyen des ISDU (voir la Figure 5).

Les Paramètres dits Directs en page 1 sont obligatoires pour tous les Dispositifs. Cette page 1 contient des paramètres communs de communication et d'identification (voir B.1).

La page 2 de Paramètres Directs offre optionnellement un espace pour au maximum 16 octets de paramètres spécifiques à la technologie (fournisseur) pour les Dispositifs qui n'exigent pas plus de données que ce nombre limité et qui ont une empreinte système faible (communication des ISDU non prise en charge, éventuel traitement des bus de terrain possible, mais moins pratique). L'accès à la page 2 de Paramètres Directs est réalisé via AL_Read et AL_Write (voir 10.7.5).

La transmission des paramètres vers et en provenance de la mémoire des index spacieuse peut être effectuée de deux manières: paramètre unique par paramètre unique ou sous forme de bloc de paramètres. La transmission par paramètre unique telle que spécifiée au 10.3.4 est assurée au moyen de plusieurs vérifications et par la confirmation du paramètre transmis. Un acquittement négatif contient une description appropriée de l'erreur et le paramètre n'est alors pas activé. La transmission sous forme de bloc de paramètres telle que spécifiée au 10.3.5 reporte la vérification de la cohérence des paramètres, ainsi que l'activation jusqu'à l'accomplissement complet de la transmission. Le Dispositif effectue les vérifications sur réception d'une commande spéciale et renvoie une confirmation ou un acquittement négatif accompagné d'une description appropriée de l'erreur. Dans ce cas, les paramètres transmis doivent être rejetés et un retour au jeu de paramètres précédent doit être effectué pour assurer une fonctionnalité correcte du Dispositif.

10.3.2 Diagramme d'états du Gestionnaire de Paramètres

Le Dispositif peut être paramétré au moyen de mécanismes d'ISDU lorsque le PM est actif. Les principales fonctions du PM sont la transmission de paramètres vers le Maître ("Téléchargement amont") et vers le Dispositif ("Téléchargement aval"), ainsi que la vérification de la cohérence et de la validité au sein du Dispositif ("Vérification de validité"), comme illustré en Figure 84.

Le PM est piloté par des messages de commande du Maître (voir le Tableau B.9). Par exemple, la protection [UploadStart] correspond à la réception de la SystemCommand (Commande Système) "ParamUploadStart" et [UploadEnd] correspond à la réception de la SystemCommand "ParamUploadEnd".

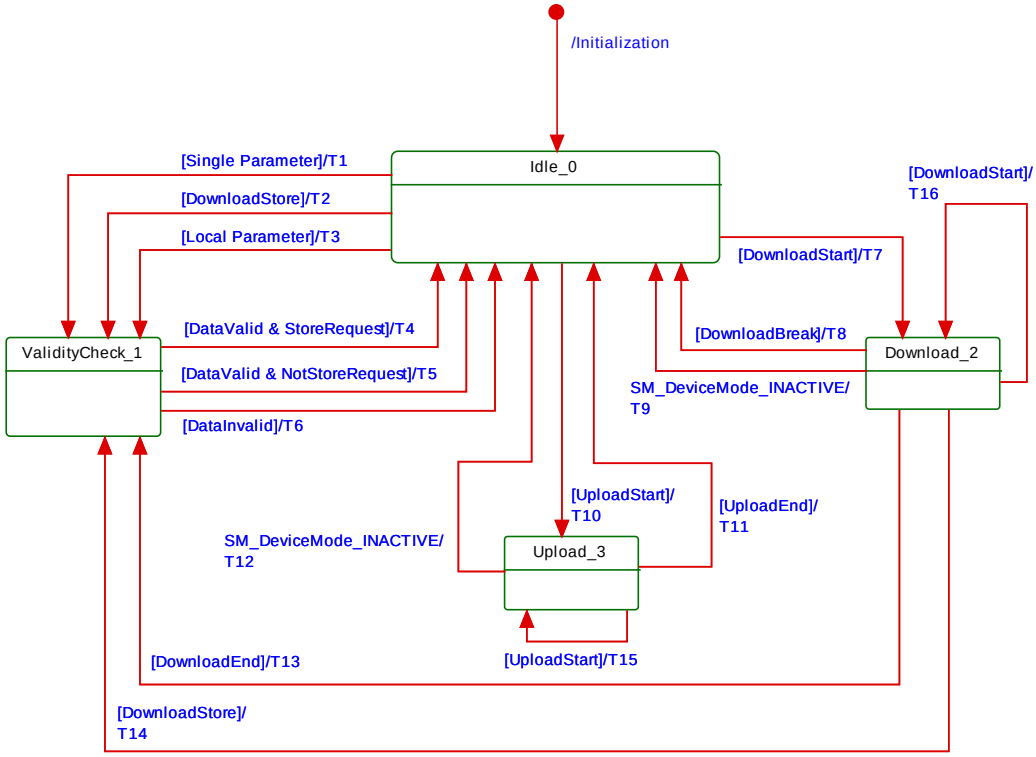
NOTE 1 Suite à une interruption de la communication, la Gestion Système du Maître utilise le service SM_DeviceMode avec la variable "INACTIVE" pour arrêter le processus de téléchargement et revenir à l'état "IDLE".

Toute nouvelle commande "ParamUploadStart" ou "ParamDownloadStart" alors qu'une autre séquence est en cours, due par exemple à l'arrêt imprévu de l'outil de paramétrage du fournisseur, entraînera l'abandon de la séquence en cours. Les modifications de paramètres correspondantes seront ignorées.

NOTE 2 Il peut y avoir conflit entre un programme utilisateur d'Automate programmable et un outil de paramétrage (accès multiples), par exemple, si au cours de la mise en service, l'utilisateur n'a pas désactivé les accès provenant du programme de l'Automate programmable pendant la modification des paramètres au moyen de l'outil.

Le mécanisme de Gestionnaire de Paramètres dans un Dispositif est toujours actif et la demande DS_ParUpload.req au cours de la transition T4 est utilisée pour déclencher le mécanisme de Stockage des Données (DS) décrit en 10.4.2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Initialization	Initialisation
Single Parameter	Paramètre unique

Figure 84 – Diagramme d'états du Gestionnaire de Paramètres (PM)

Le Tableau 94 présente les tables de transitions d'états du diagramme d'états du Gestionnaire de Paramètres (PM) du Dispositif.

Tableau 94 – Tables de transitions d'états du diagramme d'états du PM

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Idle_0	Attente de transmission de paramètres
ValidityCheck_1	Vérification de la cohérence et de la validité du jeu de paramètres courant.
Download_2	Téléchargement aval de paramètres actif; paramétrage local verrouillé (par exemple apprentissage en cours)
Upload_3	Téléchargement amont de paramètres actif; paramétrage globalement verrouillé; tous les accès en écriture pour modifier les paramètres au moyen d'outils doivent être refusés (Code d'Erreur ISDU "Service temporairement non disponible – contrôle du Dispositif")

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	0	1	Établissement de "StoreRequest" (= VRAI)
T3	0	1	Établissement de "StoreRequest" (= VRAI)
T4	1	0	Marquage du jeu de paramètres comme étant valide; Invocation de DS_ParUpload.req vers DS; activation de l'acquittement positif de la transmission; Réinitialisation de "StoreRequest" (= FAUX)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T5	1	0	Marquage du jeu de paramètres comme étant valide; activation de l'acquittement positif de la transmission
T6	1	0	Marquage du jeu de paramètres comme étant non valide; activation de l'acquittement négatif de la transmission; Réinitialisation de "StoreRequest" (= FAUX); Vidage du tampon de paramètres
T7	0	2	Verrouillage de l'accès local aux paramètres
T8	2	0	Déverrouillage de l'accès local aux paramètres; Vidage du tampon de paramètres
T9	2	0	Déverrouillage de l'accès local aux paramètres; Vidage du tampon de paramètres
T10	0	3	Verrouillage de l'accès local aux paramètres
T11	3	0	Déverrouillage de l'accès local aux paramètres
T12	3	0	Déverrouillage de l'accès local aux paramètres
T13	2	1	Déverrouillage de l'accès local aux paramètres
T14	2	1	Déverrouillage de l'accès local aux paramètres; Établissement de "StoreRequest" (= VRAI)
T15	3	3	Verrouillage de l'accès local aux paramètres
T16	2	2	Vidage du tampon de paramètres, de façon à ne pas bloquer un éventuel second démarrage.

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
DownloadStore	Bool	Commande Système "ParamDownloadStore" reçue, voir le Tableau B.9
DataValid	Bool	Résultat positif de la vérification de la conformité et de la validité
DataInvalid	Bool	Résultat négatif de la vérification de la conformité et de la validité
DownloadStart	Bool	Commande Système "ParamDownloadStart" reçue, voir le Tableau B.9
DownloadBreak	Bool	Commande Système "ParamBreak" ou "ParamUploadStart" reçue
DownloadEnd	Bool	Commande Système "ParamDownloadEnd" reçue, voir le Tableau B.9
StoreRequest	Bool	Drapeau pour une séquence de Stockage des Données demandée, c'est-à-dire Commande Système "ParamDownloadStore" reçue (= VRAI)
NotStoreRequest	Bool	Valeur inverse de StoreRequest
UploadStart	Bool	Commande Système "ParamUploadStart" reçue, voir le Tableau B.9
UploadEnd	Bool	Commande Système "ParamUploadEnd" reçue, voir le Tableau B.9
Single Parameter	Bool	En cas de "paramètre unique" comme spécifié en 10.3.4
Local Parameter	Bool	En cas de "paramètre local" comme spécifié en 10.3.3

Le Gestionnaire de Paramètres (PM) prend en charge le traitement des transferts de "paramètre unique" (Index et Sous-index) ainsi que la transmission de "paramètre de bloc" (ensemble d'un jeu de paramètres).

10.3.3 Paramètre dynamique

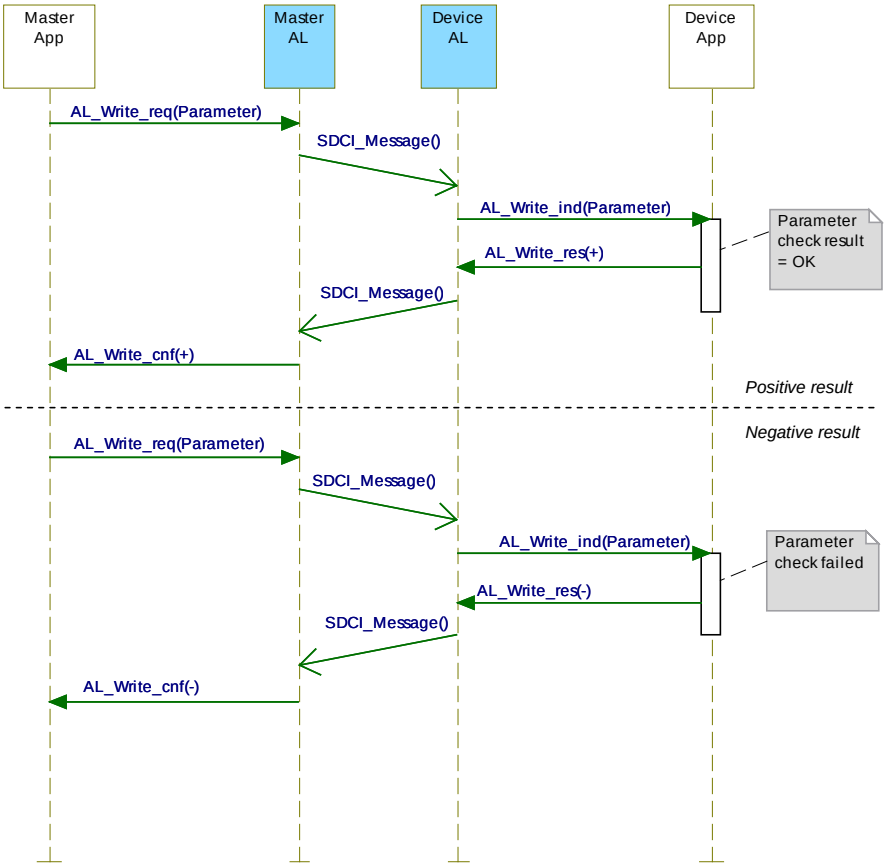
Les paramètres accessibles par le biais des services de lecture ou d'écriture SDCI peuvent également être modifiés au moyen d'éléments de contrôle embarqués (par exemple, un bouton d'apprentissage) ou de l'interface homme-machine d'un Dispositif. Ces modifications doivent subir les mêmes vérifications de validité que l'accès à un paramètre unique. Ainsi, en cas de résultat positif "DataValid" en Figure 84, le drapeau "StoreRequest" doit être appliqué afin d'assurer la cohérence du Stockage des Données. En cas de résultat négatif "InvalidData", les valeurs précédentes des paramètres correspondants doivent être restaurées ("repli"). Par ailleurs, une indication spécifique au Dispositif sur l'interface homme-machine est recommandée en tant que retour d'information positif ou négatif vers l'utilisateur.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Il est recommandé d'éviter les accès concurrents et simultanés à un paramètre via les éléments de contrôle local et les services d'écriture SDCI.

10.3.4 Paramètre unique

La Figure 85 donne des exemples d'organigrammes séquentiels pour des modifications valides et non valides de paramètre unique.



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Parameter check result = OK	Résultat de la vérification des paramètres = OK
Positive result	Résultat positif
Negative result	Résultat négatif
Parameter check failed	Échec de la vérification des paramètres

Figure 85 – Résultat positif et négatif d'une vérification de paramètre

Si un paramétrage unique est effectué via des objets ISDU, le Dispositif doit vérifier l'accès, la structure, la cohérence et la validité (voir le Tableau 95) des données transmises dans le contexte de l'ensemble du jeu de paramètres et retourner le résultat dans la confirmation. Une confirmation négative transporte une indication d'erreur du Tableau C.2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 95 – Définitions des vérifications de paramètre

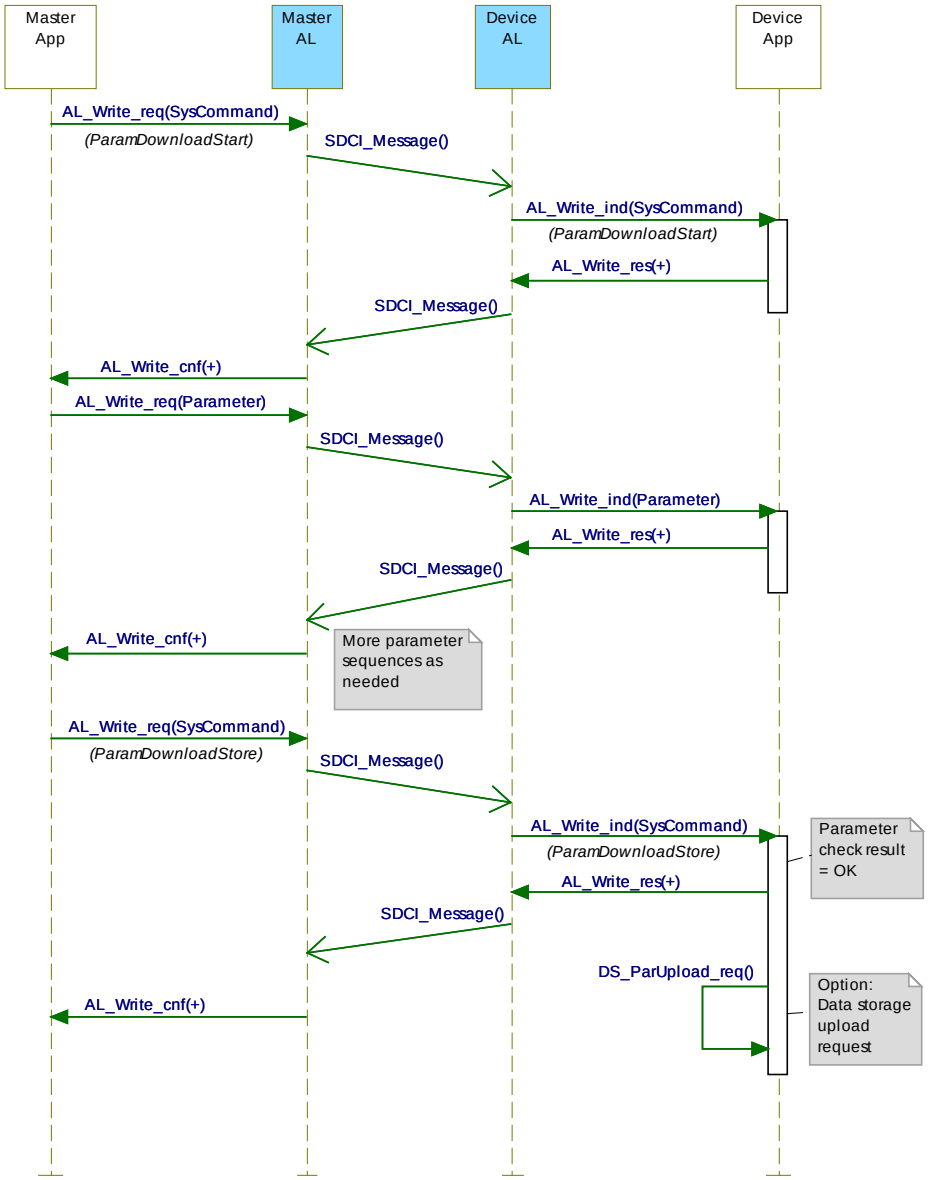
Vérification de paramètre	Définition	Indication d'erreur
Accès	Vérification des droits d'accès valides à cet Index / Sous-index, indépendamment du contenu des données (Index / Sous-index indisponible en permanence ou temporairement; accès en écriture à un Index en lecture seule)	Voir C.2.3 à C.2.8
Cohérence	Vérification du contenu valide des données de l'ensemble du jeu de paramètres, contrôle des éventuelles interférences ou corrélations entre paramètres	Voir C.2.16 et C.2.17
Structure	Vérification de la structure valide des données, telle que la taille des données, seules les structures de données complètes peuvent être écrites, par exemple 2 octets pour un type de données UInteger16	Voir C.2.12 et C.2.13
Validité	Vérification du contenu valide des données de paramètres uniques, contrôle des limites de données	Voir C.2.9 à C.2.11, C.2.14, C.2.15

10.3.5 Paramètre de bloc

Les applications utilisateur, telles que les blocs de fonctions dans les Automates programmables et les logiciels d'outils de paramétrage, peuvent utiliser les commandes de début et de fin pour indiquer le début et la fin de transmission d'un paramètre de bloc. Pendant la durée de la transmission du paramètre de bloc, l'application du Dispositif doit interdire toute modification de paramètre provenant d'autres sources, par exemple paramétrage local, apprentissage, etc.

Un exemple d'organigramme séquentiel pour des modifications valides de paramètre de bloc, avec une demande optionnelle de Stockage des Données est illustré en Figure 86.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

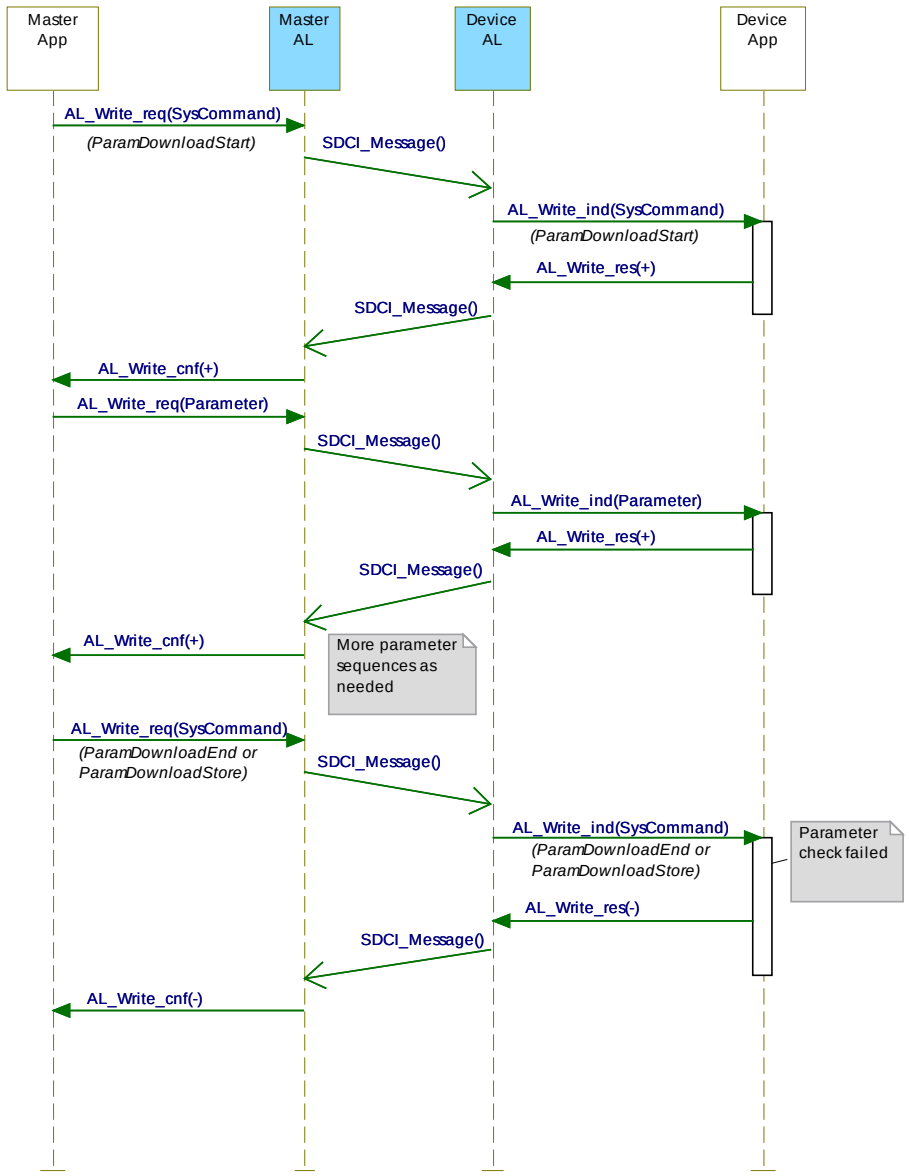
Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
(Parameter)	(paramètre)
More parameter sequences as needed	Séquences de paramètres supplémentaires, si nécessaire
Parameter check result = OK	Résultat de la vérification des paramètres = OK
Option: Data storage upload request	Option: Demande de téléchargement amont de Stockage des Données

Figure 86 – Téléchargement aval positif de paramètre de bloc avec demande de Stockage des Données

Un exemple d'organigramme séquentiel pour des modifications non valides de paramètre de bloc est illustré en Figure 87.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La commande "ParamDownloadStart" (voir le Tableau B.9) indique le début de la transmission du paramètre de bloc dans le sens de téléchargement aval (de l'application utilisateur vers le Dispositif). La SystemCommand "ParamDownloadEnd" ou "ParamDownloadStore" achève cette séquence. Les deux fonctions sont similaires. Cependant, la SystemCommand "ParamDownloadStore" force en outre le mécanisme de Stockage des Données (DS) à effectuer le téléchargement amont du jeu de paramètres par le biais de l'Événement DS_UPLOAD_REQ (voir 10.4.2).



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
(Parameter)	(paramètre)
More parameter sequences as needed	Séquences de paramètres supplémentaires, si nécessaire
Parameter check failed	Échec de la vérification des paramètres

Figure 87 – Téléchargement aval négatif de paramètre de bloc

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Pendant le téléchargement aval du paramètre de bloc, la vérification de cohérence de paramètres uniques transférés doit être neutralisée et les paramètres ne sont pas activés. Avec la commande "ParamDownloadEnd", le Dispositif vérifie l'ensemble du jeu de paramètres et indique le résultat à l'entité qui est à l'origine de la transmission du paramètre de bloc, dans l'acquittement de l'ISDU, en réponse à la commande.

Pendant le téléchargement aval du paramètre de bloc, l'accès et la structure sont toujours vérifiés (voir le Tableau 95). Il est également admis que des vérifications de la validité soient réalisées de manière facultative. Le Gestionnaire de Paramètres ne doit pas quitter le mode transfert de bloc en cas d'accès non valide ou de non-respect de la structure.

Dans le cas d'un jeu de paramètres non valide, les paramètres modifiés doivent être ignorés et un retour au jeu de paramètres précédent doit être effectué. La confirmation négative correspondante doit contenir une indication d'erreur du Tableau C.2. S'il y a confirmation négative de la SystemCommand "ParamDownloadStore", la demande de téléchargement amont de Stockage des Données est ignorée.

La commande "ParamUploadStart" (voir le Tableau B.9) indique le début de la transmission du paramètre de bloc dans le sens du téléchargement amont (depuis le Dispositif vers l'application utilisateur). La SystemCommand "ParamUploadEnd" achève cette séquence et indique la fin de la transmission.

Une transmission de paramètre de bloc est abandonnée si le Gestionnaire de Paramètres reçoit une SystemCommand "ParamBreak". Dans ce cas, la transmission de bloc est abandonnée sans aucune modification du paramétrage.

10.3.6 Accès concurrent au paramétrage

Il n'existe aucun mécanisme garantissant la cohérence des paramètres dans le Dispositif en cas d'accès simultané à partir de différentes applications d'utilisateur au-dessus du niveau du Maître. Ceci doit être garanti ou bloqué au niveau de l'utilisateur.

10.3.7 Traitement des commandes

Les commandes d'application telles que l'apprentissage ou la restauration des réglages d'usine sont acheminées sous la forme de paramètres. Une commande d'application est confirmée par une réponse positive de service – AL_Write.res(+). Une réponse négative de service – AL_Write.res(-) – doit indiquer un échec d'exécution de la commande d'application. Dans les deux cas, l'expiration du délai imparti aux ISDU doit être prise en compte (voir le Tableau 97).

10.4 Stockage des Données (DS)

10.4.1 Généralités

Le mécanisme de Stockage des Données (DS) permet une mise en mémoire tampon cohérente et actualisée des paramètres du Dispositif à des niveaux supérieurs, tels que les programmes d'Automate programmable ou le serveur de paramètre de bus de terrain. Le Stockage des Données entre Maîtres et Dispositifs est défini dans la présente norme, tandis que les mécanismes adjacents de Stockage des Données de niveau supérieur dépendent du bus de terrain ou système spécifique. Le Dispositif détient un ensemble d'objets normalisés fournissant des informations sur les paramètres pour le Stockage des Données, telles que les exigences de taille de mémoire, les informations de contrôle et d'état du mécanisme de Stockage des Données (voir le Tableau B.10). Les différentes versions des jeux de paramètres du Stockage des Données sont identifiées par une Somme de contrôle des paramètres.

Pour les Dispositifs, il est fortement recommandé de déployer le mécanisme DS défini dans la présente norme. Si ce mécanisme n'est pas pris en charge, il incombe au fournisseur du

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Dispositif de décrire la manière d'assurer convenablement et sans outillage, dans un système donné, le paramétrage d'un Dispositif après remplacement.

10.4.2 Diagramme d'états du Stockage des Données

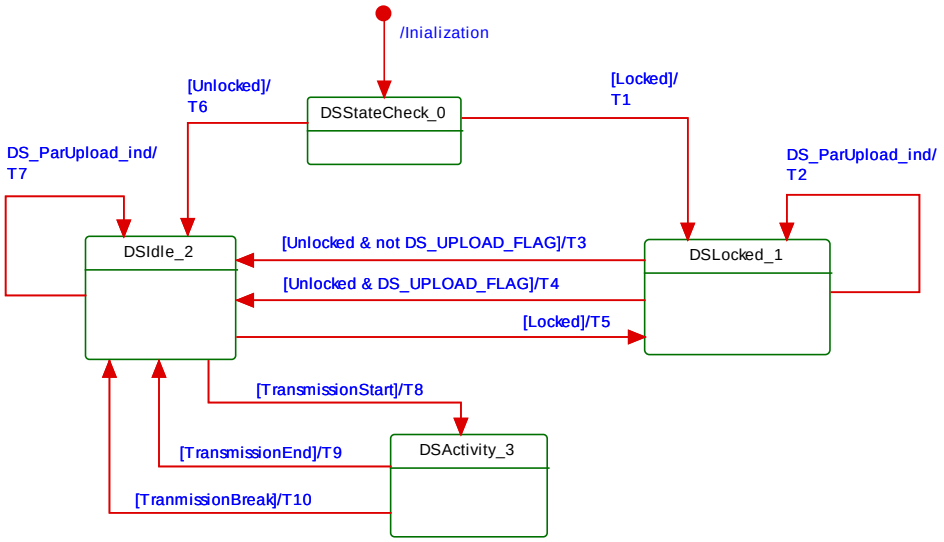
Toute modification d'un jeu de paramètres valides donne lieu à un nouveau téléchargement amont de Stockage des Données. Le téléchargement amont est lancé par le Dispositif lorsqu'il provoque un Événement "DS_UPLOAD_REQ" (voir le Tableau D.2). Le Dispositif doit stocker l'état interne "Téléchargement amont de Stockage des Données" dans une mémoire non volatile (voir le Tableau B.10, Propriété des états), jusqu'à ce qu'il reçoive une commande de Stockage des Données "DS_UploadEnd" ou "DS_DownloadEnd".

Le Dispositif doit générer un Événement "DS_UPLOAD_REQ" (voir le Tableau D.2) uniquement si le jeu de paramètres est valide et

- si les paramètres destinés au Stockage des Données ont été modifiés localement sur le Dispositif (par exemple apprentissage, interface homme-machine, etc.), ou
- si le Dispositif reçoit une SystemCommand "ParamDownloadStore".

Le mécanisme de Stockage des Données du maître est déclenché par ces informations d'Événement et lance une séquence de téléchargement amont de Stockage des Données.

Le diagramme d'états de la Figure 88 définit le mécanisme de Stockage des Données du Dispositif.



Légende

Anglais	Français
Initialization	Initialisation
Unlocked	Déverrouillé
Locked	Verrouillé
Unlocked & not DS_UPLOAD_FLAG	Déverrouillé & pas de DS_UPLOAD_FLAG
Unlocked & DS_UPLOAD_FLAG	Déverrouillé & DS_UPLOAD_FLAG

Figure 88 – Diagramme d'états du Stockage des Données (DS)

Le Tableau 96 présente les tables de transitions d'états du diagramme d'états du Stockage des Données (DS) du Dispositif. Voir le Tableau B.10 pour plus de détails concernant les attributions d'Index de Stockage des Données.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

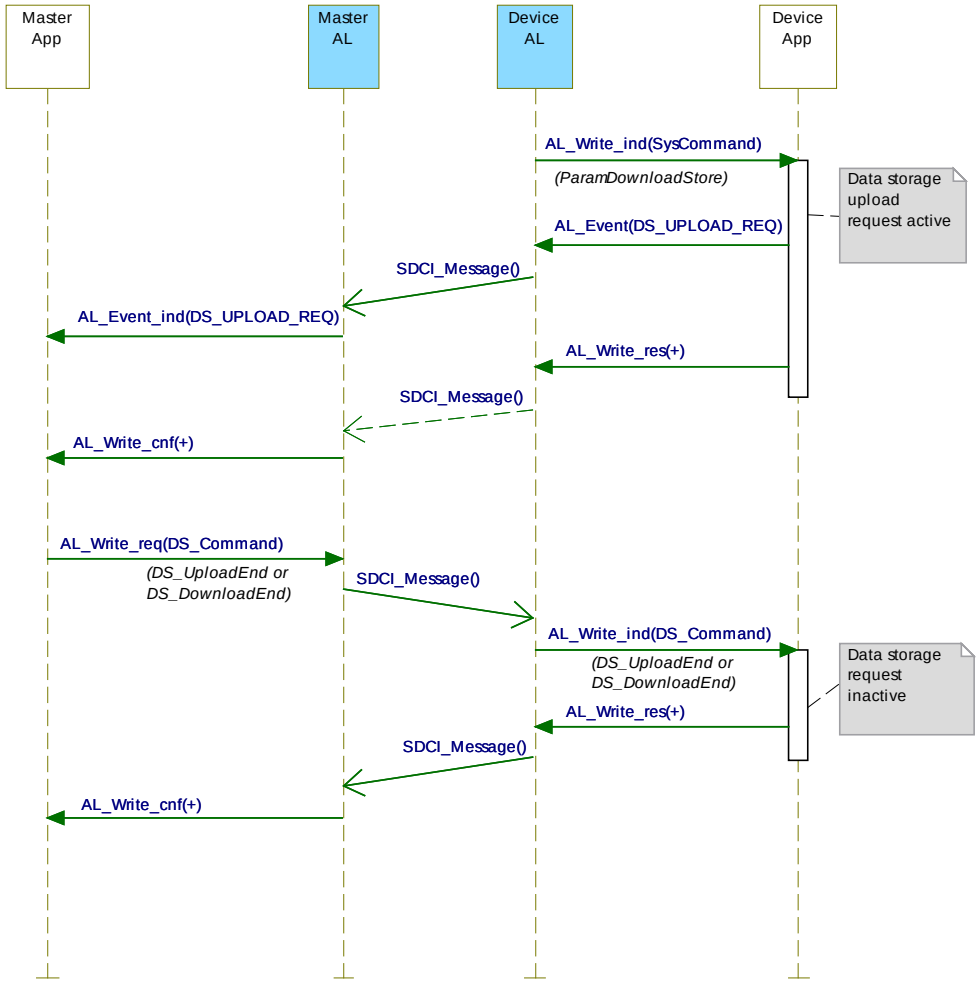
Tableau 96 – Table de transitions d'états du diagramme d'états de Stockage des Données

DÉNOMINATION DE L'ÉTAT		DESCRIPTION DE L'ÉTAT	
DSStateCheck_0		Vérification de l'état d'activation après initialisation	
DSLocked_1		Attente du déverrouillage du diagramme d'états de Stockage des Données	
DSIdle_2		Attente des activités de Stockage des données	
DSActivity_3		Fourniture du jeu de paramètres; paramétrage local verrouillé.	
TRANSITION	**ÉTAT DE LA SOURCE**	**ÉTAT DE LA CIBLE**	**ACTION**
T1	0	1	Établissement de State_Property = "accès au Stockage des Données verrouillé"
T2	1	1	Établissement de DS_UPLOAD_FLAG = VRAI
T3	1	2	Établissement de State_Property = "Inactive"
T4	1	2	Invocation de AL_EVENT.req (EventCode: DS_UPLOAD_REQ), Établissement de State_Property = "Inactive"
T5	2	1	Établissement de State_Property = "accès au Stockage des Données verrouillé"
T6	0	2	Établissement de State_Property = "Inactif"
T7	2	2	Établissement de DS_UPLOAD_FLAG = VRAI, invocation de AL_EVENT.req (EventCode: DS_UPLOAD_REQ)
T8	2	3	Verrouillage de l'accès local aux paramètres, Établissement de State_Property = "Téléchargement amont" ou "Téléchargement aval"
T9	3	2	Établissement de DS_UPLOAD_FLAG = FAUX, déverrouillage de l'accès local aux paramètres, Établissement de State_Property = "Inactif"
T10	3	2	Déverrouillage de l'accès local aux paramètres, Établissement de State_Property = "Inactif"

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
Déverrouillé	Bool	Stockage des Données déverrouillé, voir B.2.4
Verrouillé	Bool	Stockage des Données verrouillé, voir B.2.4
DS_ParUpload.ind	Service	Service interne du Dispositif entre PM et DS (voir la Figure 84)
TransmissionStart	Bool	La DS_Command "DS_UploadStart" ou "DS_DownloadStart" a été invoquée
TransmissionEnd	Bool	La DS_Command "DS_UploadEnd" ou "DS_DownloadEnd" a été invoquée
TransmissionBreak	Bool	SM_MODE_INACTIVE ou DS_Command "DS_Break" reçu

L'organigramme séquentiel tronqué de la Figure 89 illustre les séquences de communication importantes après paramétrage.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
Data storage upload request active	Demande de téléchargement amont de Stockage des Données active
Data storage request inactive	Demande de Stockage des Données inactive

Figure 89 – Séquence du message de demande de Stockage des Données

10.4.3 Configuration du DS

Le mécanisme de Stockage des Données dans le Dispositif peut être désactivé via le Maître, par exemple au moyen d'un outil ou d'un programme d'automate programmable. Voir B.2.4 pour plus de détails.

Ceci est recommandé pendant une mise en service ou des essais du système pour éviter un flux de communication trop intensif.

10.4.4 Espace mémoire du DS

Pour traiter en toutes circonstances la quantité de données à stocker, le nombre requis d'Index à sauvegarder et l'espace mémoire total requis sont fournis dans le paramètre Taille

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

de Stockage des Données; voir le Tableau B.10. L'espace mémoire total requis (y compris les informations structurelles) ne doit pas dépasser 2 048 octets (voir l'Annexe F). Le mécanisme de Stockage des Données du Maître doit pouvoir prendre en charge cette quantité de mémoire pour chaque port.

10.4.5 Index_List de DS

Le Dispositif est le "propriétaire" de la Index_List de DS (voir le Tableau B.10). Son but est de fournir les informations nécessaires au remplacement d'un Dispositif. La Index_List de DS doit être fixe pour tout DeviceID spécifique. Dans le cas contraire, l'intégrité des données entre le Maître et le Dispositif ne peut pas être garantie. La Liste des Index doit contenir le Marqueur de Fin (Termination Marker) (voir le Tableau B.10), si le Dispositif ne prend pas en charge le Stockage des Données (voir 10.4.1). Dans ce cas, la taille de stockage requise doit être 0.

10.4.6 Disponibilité des paramètres de DS

Tous les Index énumérés dans la liste doivent pouvoir être lus et écrits par les SystemCommands "DS_UploadStart" ou "DS_DownloadStart" et "DS_UploadEnd" ou "DS_DownloadEnd" (voir le Tableau B.10). Si l'un des Index est rejeté par le Dispositif, le Maître du Stockage des Données abandonnera le téléchargement (amont ou aval) par une SystemCommand "DS_Break". Dans ce cas, il n'y aura pas de nouvelle tentative de la séquence de Stockage des Données.

10.4.7 DS sans ISDU

La prise en charge de la transmission de ISDU dans un Dispositif est une condition préalable pour le Stockage des Données de paramètres. Les paramètres de la page 2 de Paramètres Directs ne peuvent pas être sauvegardés et restaurés par le mécanisme de Stockage des Données.

10.4.8 Indication de modification de paramètres de DS

La Somme de contrôle Parameter_Checksum spécifiée dans le Tableau B.10 est utilisée comme indicateur des modifications apportées à un jeu de paramètres donné. La présente norme n'exige pas un mécanisme spécifique de détection des modifications de paramètres. Un ensemble de méthodes recommandées est fourni dans l'Annexe J (informative).

10.5 Répartiteur d'Événements (ED)

Toute application du Dispositif peut générer des informations prédéfinies d'état système lorsque les opérations de l'interface SDCI échouent ou lorsqu'il apparaît des informations spécifiques à la technologie (diagnostic) suite à des méthodes de diagnostic spécifiques à la technologie. Le Répartiteur d'Événements transforme ces informations en Événement conformément aux définitions données en A.6. L'Événement consiste en un EventQualifier (Qualificatif d'Événement) indiquant les propriétés d'un incident et en un EventCode ID (Identifiant de Code d'Événement) qui est un descriptif de l'incident, accompagné des éventuelles mesures correctives. Le Tableau D.1 donne une liste des identifiants et descriptifs prédéfinis pour des incidents orientés application. Des plages d'identifiants sont réservées aux incidents spécifiques au profil et au fournisseur. Le Tableau D.2 donne une liste des identifiants prédéfinis pour des incidents spécifiques à l'interface SDCI.

Les Événements sont classés en "Erreurs", "Avertissements" et "Notifications". Le paragraphe 10.9.2 fournit des recommandations sur la manière d'affecter ces classifications; la méthode utilisée par le Maître pour contrôler et traiter ces Événements est donnée en 11.5.

Tous les Événements fournis à un moment donné dans le temps sont acquittés par une commande unique. Par conséquent, l'acquittement de l'Événement peut être retardé par l'acquittement le plus lent en provenance des niveaux système supérieurs.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.6 Fonctionnalités du Dispositif

10.6.1 Généralités

Les fonctionnalités de Dispositif suivantes sont définies jusqu'à un certain point pour obtenir un comportement commun. Elles sont accessibles par les méthodes ou paramètres normalisés ou spécifiques au Dispositif. La disponibilité de ces fonctionnalités est définie dans l'IODD d'un Dispositif.

10.6.2 Rétrocompatibilité d'un Dispositif

Cette fonctionnalité permet à un Dispositif de jouer le rôle d'une version de Dispositif précédente. Au cours de la phase de démarrage, la Gestion Système du Maître écrase l'identifiant de Dispositif (DID) inhérent au Dispositif et le remplace par l'ancien DeviceID requis. L'application spécifique à la technologie du Dispositif doit basculer sur les anciens ensembles ou sous-ensembles fonctionnels attribués à ce DeviceID. La prise en charge de la rétrocompatibilité est facultative pour un Dispositif donné.

Étant donné qu'un Dispositif peut assurer une rétrocompatibilité avec des DeviceID (DID) précédents, ces Dispositifs compatibles doivent prendre en charge tous les paramètres et capacités de communication de l'identifiant de Dispositif précédent. Ainsi, le Dispositif est autorisé dans ce cas à modifier tout paramètre de communication ou d'identification.

10.6.3 Compatibilité de versions du protocole

Cette fonctionnalité permet à un Dispositif d'ajuster ses couches protocolaires sur une version précédente du protocole SDCI, telle que par exemple, sur la version de protocole héritage d'un Maître héritage ou à l'avenir, de la version V(x) à la version V(x-n). Au cours de la phase de démarrage, la Gestion Système du Maître peut écraser le RevisionID (ID de version) (RID) inhérent au protocole du Dispositif en cas de divergence avec le RevisionID pris en charge par le Maître. Un Maître héritage n'écrit pas la MasterCommand "MasterIdent" (voir le Tableau B.2) et le Dispositif peut ainsi s'adapter au protocole héritage (V1.0). La prise en charge de la compatibilité entre versions est facultative pour un Dispositif donné.

10.6.4 Réglages d'usine

Cette fonctionnalité permet à un Dispositif de restaurer les paramètres à leur état initial au moment de la livraison. Le drapeau de Stockage des Données et d'autres paramètres dynamiques tels que "Nombre d'erreurs" (voir B.2.17), "État du Dispositif" (voir B.2.18) et "État détaillé du Dispositif" (voir B.2.19) doivent être réinitialisés lorsque cette fonctionnalité est appliquée. Ceci ne comprend pas les paramètres spécifiques au fournisseur, comme par exemple les compteurs d'heures de fonctionnement.

NOTE Dans ce cas, un jeu existant de paramètres enregistré dans le Maître sera automatiquement téléchargé dans le Dispositif après démarrage.

Il incombe au fournisseur de garantir le fonctionnement correct en toutes circonstances. La réinitialisation est déclenchée par la réception de la SystemCommand "Restauration des réglages d'usine" (voir le Tableau B.9). La réinitialisation en tant que retour aux réglages d'usine est facultative pour un Dispositif donné.

10.6.5 Réinitialisation d'application

Cette fonctionnalité permet de réinitialiser l'application spécifique à la technologie d'un Dispositif donné. Elle est notamment utile lorsqu'une application spécifique à la technologie est à configurer sur un état opérationnel prédéfini sans interruption de la communication et sans cycle d'arrêt. La réinitialisation est déclenchée par la réception d'une SystemCommand "Réinitialisation d'application" (voir le Tableau B.9). La réinitialisation de l'application spécifique à la technologie est facultative pour un Dispositif donné.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.6.6 Réinitialisation d'un Dispositif

Cette fonctionnalité permet à un Dispositif d'effectuer un "démarrage à chaud". Elle est notamment utile lorsqu'il faut réinitialiser un Dispositif à un état initial, comme lors de la mise sous tension. Dans ce cas, la communication sera interrompue. Le démarrage à chaud est déclenché par la réception d'une SystemCommand "Réinitialisation d'un Dispositif" (voir le Tableau B.9). Le démarrage à chaud est facultatif pour un Dispositif donné.

10.6.7 Indication visuelle d'état de l'interface SDCI

Cette fonctionnalité indique l'état opérationnel de l'interface SDCI du Dispositif. L'indication du mode SDCI est spécifiée en 10.9.3. L'indication du mode SIO est spécifique au fournisseur et n'est pas couverte par la présente définition. La fonction est déclenchée par une indication de la Gestion Système (dans tous les états sauf SM_Idle et SM_SIO en Figure 79). L'indication de l'état SDCI est facultative pour un Dispositif donné.

10.6.8 Verrouillage de l'accès aux paramètres

Cette fonctionnalité permet à un Dispositif de verrouiller ou de déverrouiller globalement l'accès en écriture à tous les paramètres modifiables du Dispositif accessible via l'interface SDCI (voir B.2.4). Le verrouillage est déclenché par la réception d'un paramètre système "Verrouillage d'accès au Dispositif" (voir le Tableau B.8). La prise en charge de ces fonctions est facultative pour un Dispositif donné.

10.6.9 Verrouillage du Stockage des Données

L'établissement de ce verrou fera passer "State_Property" dans le Tableau B.10 à l'état "Stockage des Données verrouillé" et dans ce cas, le Dispositif n'envoie pas un Événement DS_UPLOAD_REQ. La prise en charge de cette fonction est obligatoire pour un Dispositif donné si le mécanisme de Stockage des Données est utilisé.

10.6.10 Verrouillage de paramètres du Dispositif

Ce verrouillage désactivera l'écrasement ou la modification de paramètres du Dispositif par des contrôles embarqués ou des éléments de réglage, tels que des boutons d'apprentissage (voir B.2.4). La prise en charge de cette fonction est facultative pour un Dispositif donné.

10.6.11 Verrouillage de l'interface utilisateur d'un Dispositif

Ce verrouillage désactivera le fonctionnement des affichages et éléments de réglage de l'interface homme-machine embarquée, tels que les boutons d'apprentissage, sur un Dispositif donné (voir B.2.4). La prise en charge de cette fonction est facultative pour un Dispositif donné.

10.6.12 Décalage temporel

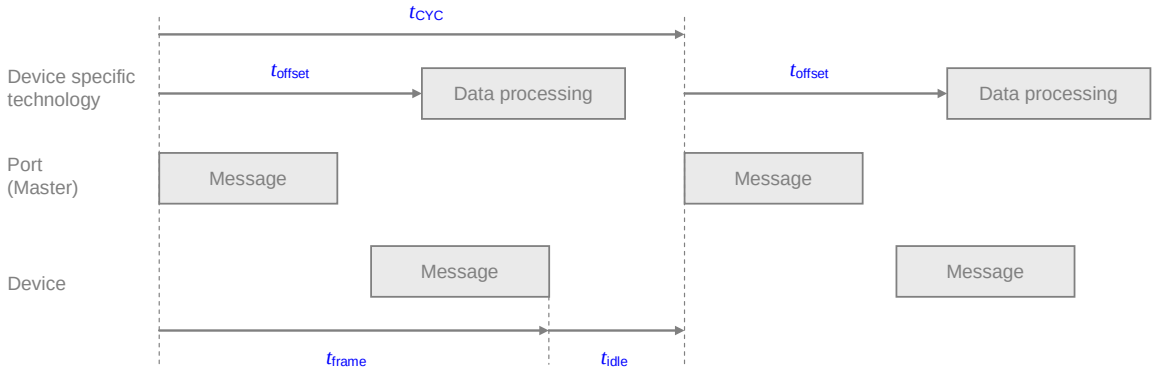
Le décalage temporel t_{offset} est un paramètre à configurer par l'utilisateur (voir B.2.22). Il détermine le début du traitement des données spécifiques à la technologie du Dispositif par rapport au début du cycle de séquence M, c'est-à-dire le début du message de (port) Maître.

Ce décalage permet:

- de synchroniser le traitement des données d'un Dispositif avec le cycle du (port) Maître dans certaines limites;
- de synchroniser entre eux les traitements de données de plusieurs Dispositifs sur différents ports du Maître;
- d'exécuter avec un décalage défini le traitement de données de plusieurs Dispositifs sur différents ports du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Figure 90 représente la synchronisation des messages par rapport au traitement des données dans les Dispositifs.



Légende

Anglais	Français
Device specific technology	Technologie spécifique au Dispositif
Data processing	Traitement des données
Port (Master)	Port (Maître)
Message	Message
Device	Dispositif

Figure 90 – Synchronisation du cycle

Le décalage temporel définit un déclencheur par rapport au début d'un cycle de séquence M. La prise en charge de cette fonction est facultative pour un Dispositif donné.

10.6.13 Concept de Stockage des Données

Le mécanisme de Stockage des Données dans un Dispositif permet de sauvegarder automatiquement des paramètres dans le serveur de Stockage des Données du Maître et de les restaurer sur notification d'Événements. La cohérence des données est vérifiée dans chaque sens dans le Maître et dans le Dispositif. Le Stockage des Données concerne principalement les paramètres de configuration d'un Dispositif réglé lors de la mise en service (voir 10.4 et 11.3). La prise en charge de cette fonction est facultative pour un Dispositif donné.

10.6.14 Paramètre de bloc

La fonctionnalité de transmission de paramètre de bloc dans un Dispositif permet le transfert de jeux de paramètres à partir d'un programme d'Automate programmable sans effectuer de vérification de la cohérence de chaque objet de données unique. La vérification de la validité et de la cohérence est réalisée à la fin de la transmission du paramètre de bloc pour l'ensemble du jeu de paramètres. Cette fonction concerne principalement l'échange de paramètres d'un Dispositif à configurer dans l'environnement d'exécution (voir 10.3). La prise en charge de cette fonction est facultative pour un Dispositif donné.

10.7 Règles et contraintes de conception d'un Dispositif

10.7.1 Généralités

Outre les définitions du protocole sous forme d'états, de séquences, d'activités et de chronogrammes, un certain nombre de règles et de contraintes supplémentaires sont nécessaires pour définir le comportement des Dispositifs. Un aperçu des principales variables

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

du protocole, réparties dans l'ensemble de la norme, est regroupé dans le Tableau 97, qui fournit également les références correspondantes.

10.7.2 Données de Processus

Le canal de communication du processus transmet de manière cyclique les Données de Processus sans d'éventuelles interférences des canaux de communication des Données à la Demande. L'échange de Données de Processus commence automatiquement dès que le Dispositif passe à l'état OPERATE via un message provenant du Maître.

Le format de données transmises est spécifique au Dispositif et varie de zéro à 32 octets de données dans chaque sens de communication.

Recommandations.

- Il convient que les structures de données conviennent à une utilisation par des applications d'Automate programmable.
- Il est fortement recommandé de satisfaire aux règles données en E.3.3 et dans la référence [6].

Pour plus de détails concernant l'indication de Données de Processus valides ou non valides, via un drapeau PDValid, au cours de l'échange cyclique de données, voir A.1.5.

10.7.3 Perte de communication

Il incombe au concepteur du Dispositif de définir le comportement convenable du Dispositif en cas de perte de communication avec le Maître (la transition T10 en Figure 42 gère la détection de la perte de communication et les actions du Dispositif qui en résultent sont définies en 10.2).

NOTE Ceci est notamment important pour des actionneurs tels que les vannes ou les systèmes de gestion de moteurs.

10.7.4 Paramètre direct

La communication avec la page de Paramètres Directs ne comporte pas de mécanismes d'échange d'authentification (poignée de main) pour assurer une réception correcte ou la validité des paramètres transmis. Il n'est possible d'accéder à la page de Paramètres Directs que par un octet unique à la fois (Sous-index) ou par ensemble (16 octets). Par conséquent, la cohérence de paramètres d'une taille supérieure à 1 octet ne peut être garantie en cas d'accès par octet unique.

Les paramètres de la page de Paramètres Directs ne peuvent pas être sauvegardés et restaurés via le mécanisme de Stockage des Données.

10.7.5 Canal de communication des ISDU

Le canal de communication des ISDU est un moyen puissant de transmission de paramètres et de commandes (voir B.2).

Les règles suivantes doivent être prises en compte lorsque ce canal est utilisé (voir la Figure 6).

- L'Index 0 n'est pas accessible via le canal de communication des ISDU. L'accès est redirigé par le Maître vers la page 1 de Paramètres Directs en utilisant le canal de communication de page.
- L'Index 1 n'est pas accessible via le canal de communication des ISDU. L'accès est redirigé par le Maître vers la page 2 de Paramètres Directs en utilisant le canal de communication de page.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- Un programme d'application d'Automate programmable ne peut pas accéder à l'Index 3. L'accès est limité à l'application du Maître uniquement (Stockage des Données).
- Après réception d'une demande de ISDU provenant du Maître, le Dispositif doit répondre dans un délai de 5 000 ms (voir le Tableau 97). Toute infraction à cette règle entraîne l'abandon de la tâche en cours par le Maître.

10.7.6 Règles de DeviceID relatives aux variantes de Dispositifs

Les Dispositifs ayant un certain DeviceID et un certain VendorID ne doivent pas déroger au comportement de communication et de fonctionnement. Ceci s'applique aux capteurs et aux actionneurs. Les différences entre ces Dispositifs peuvent être par exemple

- les longueurs de câblage,
- les matériaux des enveloppes,
- les mécanismes de montage,
- d'autres fonctionnalités et conditions environnementales.

10.7.7 Constantes de protocole

Le Tableau 97 donne un aperçu des principales constantes de protocole pour les Dispositifs.

Tableau 97 – Aperçu des constantes de protocole pour les Dispositifs

Variable Système	Références	Valeurs	Définition
Délai d'acquittement des ISDU, par exemple après une Commande Système	B.2.2	5 000 ms	Délai à compter de la réception d'une ISDU, par exemple une Commande Système et le début du message de réponse du Dispositif (voir la Figure 61)
Nombre maximal d'entrées dans la Liste des Index	B.2.3	70	Chaque entrée comporte un Index et un Sous-index. 70 entrées donnent au total 210 octets.
Des valeurs de préréglage pour des paramètres non utilisés ou réservés, par exemple FunctionID	Annexe B	0 (s'il s'agit de chiffres) 0x00 (s'il s'agit de caractères)	L'ingénierie doit mettre tous les paramètres non utilisés aux valeurs de préréglage.
Procédure de réactivation	7.3.2.2	Voir le Tableau 40 et le Tableau 41	Temporisations minimales et maximales et nombre de nouvelles tentatives
MaxRetry	7.3.3.3	2, voir le Tableau 44	Nombre maximal de nouvelles tentatives après des erreurs de communication
MinCycleTime	A.3.7 et B.1.3	Voir le Tableau A.11 et le Tableau B.3	Le Dispositif définit sa durée de cycle minimale pour l'acquisition de données de processus d'entrée ou de sortie.
Plage d'Index utilisable	B.2	Voir le Tableau B.8	La présente version de la norme réserve certaines zones dans la plage totale de 65 535 index.
Erreurs et avertissements	10.9.2	50 ms	Un Événement de MODE "Apparition d'Événement" doit au moins persister pendant cette durée.
EventCount	8.2.2.11	1	Contrainte pour AL_Event.req

10.8 Descriptif de Dispositif d'E/S (IODD)

Un IODD (Descriptif de Dispositif d'E/S) est un fichier qui fournit l'ensemble des propriétés nécessaires à l'établissement de la communication ainsi que les paramètres et limites correspondantes nécessaires à l'établissement de la fonction souhaitée d'un capteur ou d'un actionneur.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Un IODD (Descriptif de Dispositif d'E/S) est un fichier qui décrit formellement un Dispositif.

Un fichier IODD doit être fourni pour chaque Dispositif et inclure toutes les informations nécessaires à l'appui de la présente norme.

L'IODD peut être utilisé par les outils techniques d'ingénierie d'Automates programmables et/ou de Maîtres pour l'identification, la configuration, la définition des structures de données d'Échange de Données de Processus, le paramétrage et le décodage de diagnostic d'un Dispositif particulier.

NOTE Une description détaillée utilisée par le langage IODD pour décrire un Dispositif est donnée dans la référence [6].

10.9 Diagnostic d'un Dispositif

10.9.1 Concepts

À l'Article D.2, la présente norme fournit uniquement les EventCodes (codes d'événements) les plus communs. L'objectif de ces informations de diagnostic communes est de permettre à un opérateur ou à un agent de maintenance de mettre rapidement en œuvre des mesures correctives sans avoir une connaissance approfondie de la technologie du Dispositif. Ainsi, le texte associé à un Code d'Événement particulier doit toujours contenir une instruction de maintenance corrective accompagnant les informations de diagnostic.

Les passerelles de Maître de bus de terrain ont tendance à n'établir de correspondance que pour quelques EventCodes jusqu'au niveau de système supérieur. Généralement, les EventCodes spécifiques au fournisseur définis par l'IODD peuvent uniquement être décodés en instructions lisibles via un Outil de configuration des ports et Dispositifs (PDCT) ou un outil spécifique au fournisseur utilisant l'IODD.

Des informations condensées relatives à l'"état de santé" du Dispositif peuvent être récupérées à partir du paramètre "État du Dispositif" (voir B.2.18). Le Tableau 98 donne un aperçu des diverses possibilités offertes aux Dispositifs et présente des exemples de consommateurs de ces informations.

Si cette fonction est mise en œuvre, il est également possible de lire le nombre de défauts depuis la mise sous tension ou la réinitialisation via le paramètre "Nombre d'erreurs" (voir B.2.17) et des informations supplémentaires dans le cas de profils de Dispositifs via le paramètre "État détaillé du Dispositif" (voir B.2.19).

NOTE Les valeurs spécifiques aux profils pour l'"État détaillé du Dispositif" sont données dans la référence [7].

Si nécessaire, il est fortement recommandé de fournir des informations supplémentaires de diagnostic "plus approfondies" et spécifiques à la technologie, sous la forme de paramètres spécifiques au Dispositif (voir le Tableau B.8) qui peuvent être récupérés via des Outils de configuration des ports et Dispositifs pour des Maîtres ou via des outils spécifiques au fournisseur. Généralement, seuls des spécialistes ou le personnel d'entretien du fournisseur sont en mesure de tirer des conclusions à partir de ces informations.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 98 – Classification d'incidents de diagnostiques de Dispositifs

Incident diagnostiqué	Apparition/ disparition	Occurrence unique	Paramètre	Destination	Consommateur
Erreur (mesure corrective rapide; Codes d'événements normalisés)	oui	-	-	Automate programmable ou IHM (Mise en correspondance du bus de terrain)	Personnel chargé de la maintenance et des réparations
Erreur (IODD: Codes d'événements spécifiques au fournisseur; voir le Tableau D.1)	oui	-	-	PDCT ou outillage du fournisseur	Personnel d'entretien du fournisseur
Erreur (via des paramètres spécifiques au Dispositif)	-	-	Voir le Tableau B.8	PDCT ou outillage du fournisseur	Personnel d'entretien du fournisseur
Avertissement (mesure corrective rapide; Codes d'événements normalisés)	oui	-	-	Automate programmable ou IHM	Personnel chargé de la maintenance et des réparations
Avertissement (IODD: Codes d'événements spécifiques au fournisseur; voir le Tableau D.1)	oui	-		PDCT ou outillage du fournisseur	Personnel d'entretien du fournisseur
Avertissement (via des paramètres spécifiques au Dispositif)	-	-	Voir le Tableau B.8		
Notification (Codes d'événements normalisés)	-	oui		PDCT	Personnel chargé de la mise en service
État détaillé du Dispositif	-	-		PDCT ou outillage du fournisseur	Personnel chargé de la mise en service et personnel d'entretien du fournisseur
Nombre de défauts via le paramètre "Nombre d'erreurs"	-	-	Voir B.2.17		
"État de santé" du Dispositif via le paramètre "État du Dispositif"	-	-	Voir B.2.18, le Tableau B.13	IHM, des outils tels que "Gestion d'actifs"	Opérateur

10.9.2 Événements

Les valeurs de MODE doivent être attribuées comme suit (voir A.6.4):

- Les Événements de TYPE "Erreur" doivent utiliser les MODES "Apparition / Disparition d'Événement";
- Les Événements de TYPE "Avertissement" doivent utiliser les MODES "Apparition / Disparition d'Événement";
- Les Événements de TYPE "Notification" doivent utiliser le MODE "occurrence unique d'Événement".

Les exigences suivantes sont applicables:

- Tous les Événements déjà placés dans la file d'attente des Événements sont ignorés par le Répartiteur d'Événements lorsque la communication est interrompue ou annulée.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

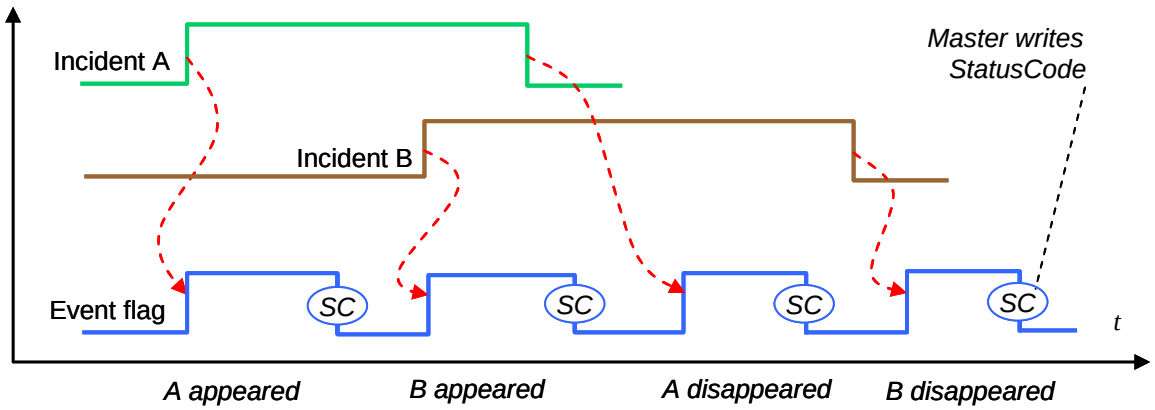
NOTE Après reprise de la communication, l'application spécifique à la technologie est chargée de rendre correctement compte des causes de l'Événement en cours.

- Il incombe au Répartiteur d'Événements de contrôler le flux d''Apparition d'Événement" et de "Disparition d'Événement". Une fois que le Répartiteur d'Événements a transmis un Événement en MODE "Apparition d'Événement" pour un EventCode donné, il ne doit pas le transmettre une nouvelle fois pour le même EventCode, avant d'avoir transmis un Événement en MODE "Disparition d'Événement" pour ce même EventCode.
- Chaque Événement doit utiliser un mode statique, le type et les attributs d'instance.
- Chaque EventCode spécifique au fournisseur doit être attribué de manière unique à l'un des TYPES (Erreur, Avertissement ou Notification).

Pour prévenir une surcharge du canal de communication de diagnostic (voir la Figure 6), les exigences suivantes s'appliquent:

- Les mêmes informations de diagnostic ne doivent pas être signalées à des intervalles de moins de 60 s, c'est-à-dire que le Répartiteur d'Événements ne doit pas invoquer le service AL_Event avec le même EventCode dans un intervalle de moins de 60 s.
- Le Répartiteur d'Événements ne doit pas produire de MODE "Disparition d'Evénement" dans un intervalle de moins de 50 ms après l'occurrence du MODE "Apparition d'Événement" correspondant.
- Les occurrences ultérieures d'erreurs ou d'avertissements ayant la même cause première doivent être ignorées; en d'autres termes, une cause première doit donner lieu à une seule indication d'erreur ou d'avertissement.
- Le Répartiteur d'Événements ne doit pas invoquer le service AL_Event avec un EventCount supérieur à un.
- Les erreurs sont prioritaires par rapport aux avertissements.

La Figure 91 illustre la manière dont deux erreurs successives sont traitées et le flux correspondant d'"Apparition / Disparition d'Événement" pour chaque erreur.



Légende

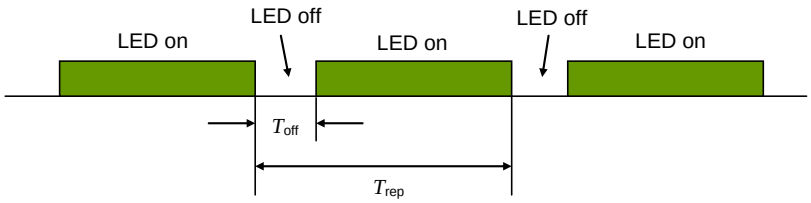
Anglais	Français
Incident	Incident
Master writes Status Code	Le Maître écrit le "Code d'État"
Event flag	Drapeau d'Événement
A appeared	Apparition de A
B appeared	Apparition de B
A disappeared	Disparition de A
B disappeared	Disparition de B

Figure 91 – Flux d'Événements en cas d'erreurs successives

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

10.9.3 Indicateurs visuels

L'indication de la communication SDCI sur le Dispositif est facultative. L'indication SDCI doit utiliser un indicateur de couleur verte. L'indication suit la temporisation et la spécification, comme illustré en Figure 92.



Légende

Anglais	Français
LED off	DEL éteinte
LED on	DEL allumée

Figure 92 – Temporisation de l'indicateur à DEL du Dispositif

Le Tableau 99 définit la temporisation de l'indicateur à DEL des Dispositifs.

Tableau 99 – Temporisation pour des indicateurs à DEL

Temporisation	Minimale	Type	Maximale	Unité
t_{rep}	750	1 000	1 250	ms
T_{off}	75	100	150	ms
T_{off} /T_{rep}	7,5	10	12,5	%

NOTE Les temporisations ci-dessus sont définies de façon à ce que la perception générale soit "dispositif sous tension".

Une courte interruption périodique indique que le Dispositif est à l'état de communication COMx. Pour éviter le phénomène de clignotement, le cycle d'indication doit commencer par l'état "DEL éteinte" et doit toujours s'achever (voir le Tableau 99).

10.10 Connectivité d'un Dispositif

Les différentes possibilités de connexion de Dispositifs aux ports du Maître ainsi que les types de câbles et les codes de couleur correspondants, sont donnés en 5.5.

NOTE Pour des raisons de compatibilité, la présente norme n'interdit pas l'utilisation de conducteurs supplémentaires sur des Dispositifs SDCI pour la connexion des fonctions non couvertes par le domaine d'application de la présente norme (par exemple pour le transfert de signaux de sortie analogiques).

11 Maître

11.1 Aperçu

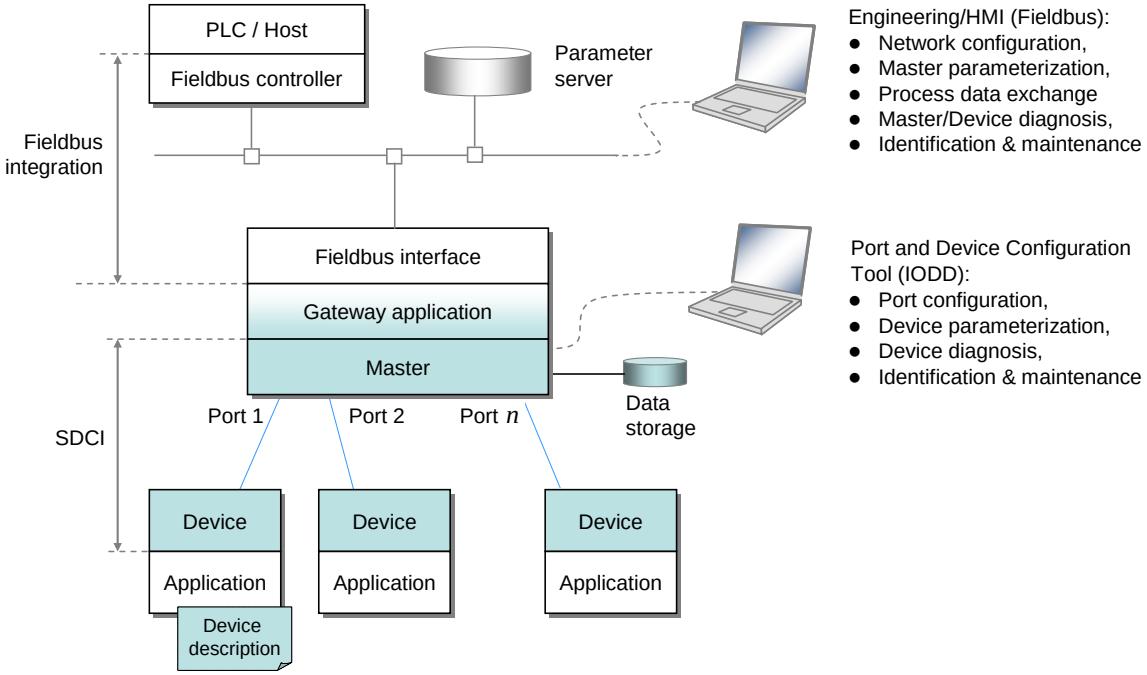
11.1.1 Modèle générique d'intégration système d'un Maître

Le domaine de la technologie SDCI dans le cadre de la hiérarchie de l'automatisation a déjà été présenté en 4.2.

La Figure 93 illustre la relation recommandée entre technologie SDCI et technologie de bus de terrain. Même s'il peut s'agir du cas d'utilisation le plus important dans la pratique, ceci

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

n'implique pas automatiquement une dépendance de la technologie SDCI de son intégration dans les systèmes de bus de terrain. L'interface SDCI peut également être directement intégrée dans les systèmes d'Automate programmable, dans les ordinateurs industriels ou dans d'autres systèmes de commande qui ne sont pas reliés entre eux par une communication de bus de terrain.



Légende

Anglais	Français
PLC / Host	Automate programmable / hôte
Fieldbus controller	Contrôleur de bus de terrain
Parameter server	Serveur de paramètres
Fieldbus integration	Intégration au bus de terrain
Fieldbus interface	Interface de bus de terrain
Gateway application	Application passerelle
Master	Maître
Data storage	Stockage des données
Device Application	Application du Dispositif
Device description	Description de dispositif
Engineering/HMI (Fieldbus):	Technique/IHM (bus de terrain):
Network configuration,	Configuration du réseau,
Master parameterization,	Paramétrage du Maître,
Process data exchange	Échange de Données de Processus,
Master/Device diagnosis,	Diagnostic Maître/Dispositif,
Identification & maintenance	Identification & maintenance
Port and Device Configuration Tool (IODD):	Outil de configuration des ports et Dispositifs (IODD):
Port configuration,	Configuration des Ports,
Device parameterization,	Paramétrage du Dispositif,
Device diagnosis,	Diagnostic du Dispositif,
Identification & maintenance	Identification & maintenance

NOTE Les zones bleutées indiquent les fonctionnalités spécifiées dans la présente norme.

Figure 93 – Relation générique entre technologie SDCI et technologie de bus de terrain

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.1.2 Structure et services d'un Maître

La Figure 94 donne un aperçu de l'ensemble de la structure et des services d'un Maître.

Les applications de Maître comportent en premier lieu une passerelle spécifique au bus de terrain ou une connexion directe à un Automate programmable (hôte) permettant d'effectuer la configuration de démarrage et le paramétrage ainsi que les échanges de Données de Processus, la modification des paramètres contrôlés par le programme de l'utilisateur dans l'environnement d'exécution et la propagation des informations de diagnostic. Pour la configuration, le paramétrage et le diagnostic au cours de la mise en service, un composant logiciel appelé "Outil de configuration des ports et Dispositifs" est connecté directement au Maître ou via la communication de bus de terrain. Ces deux instruments utilisent les applications communes du Maître décrites ci-après:

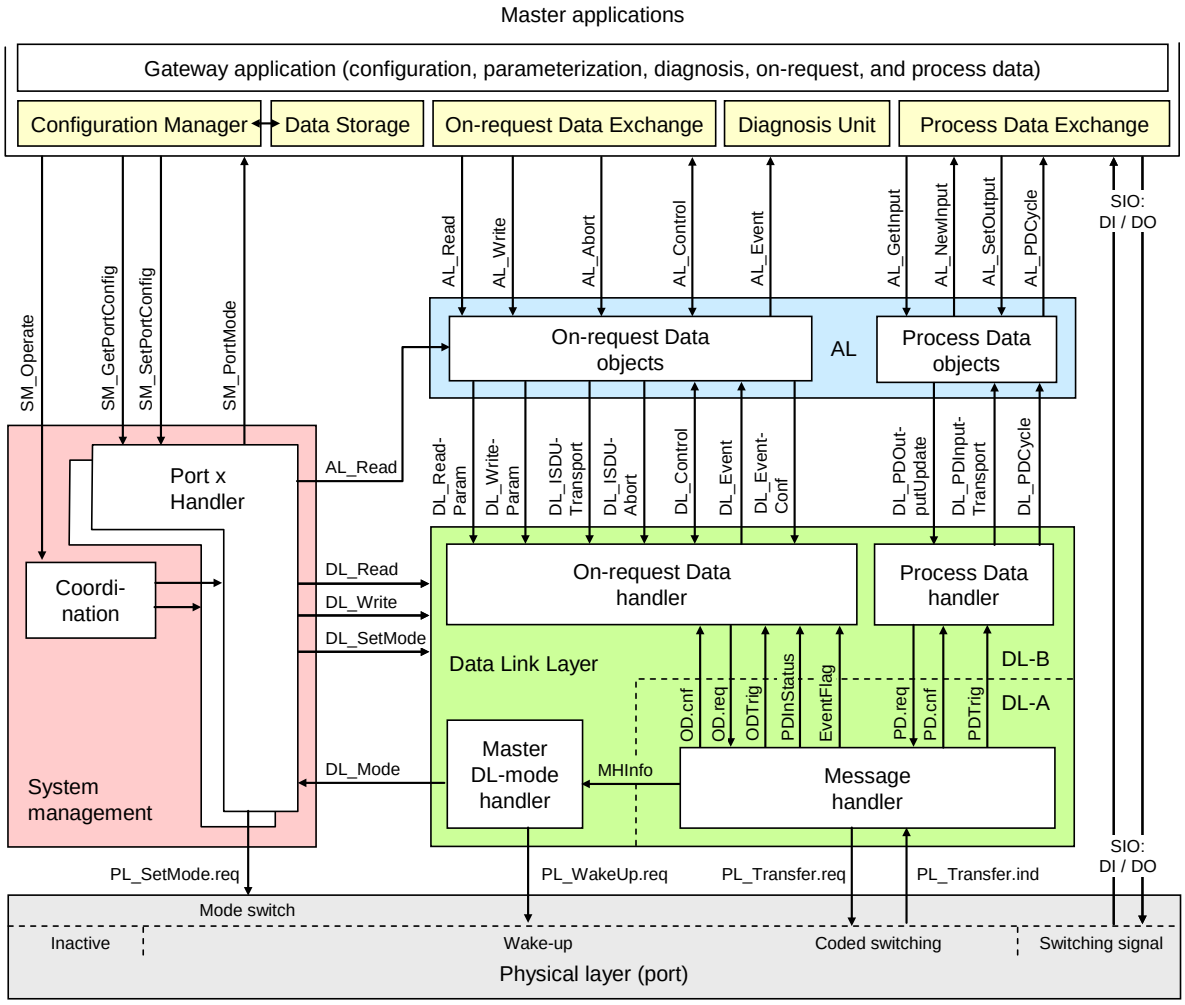
- le Gestionnaire de Configuration (CM), qui transforme les attributions de configuration de l'utilisateur en réglages de ports;
- l'Échange de Données à la Demande (ODE), qui fournit par exemple l'accès acyclique aux paramètres;
- le mécanisme de Stockage des Données (DS), qui peut être utilisé pour sauvegarder et restaurer les paramètres d'un Dispositif;
- l'Unité de Diagnostic (DU), qui achemine les Événements de la couche AL à l'unité de Stockage des Données ou à l'application passerelle;
- l'Échange de Données de Processus (PDE), qui constitue le pont de liaison avec les instruments d'automatisation de niveau supérieur.

Ces applications du Maître constituent des méthodes/fonctions normalisées communes à tous les Maîtres.

Le Gestionnaire de Configuration (CM) et le mécanisme de Stockage des Données (DS) nécessitent une coordination particulière vis-à-vis des Données à la Demande (voir la Figure 95 et la Figure 105).

L'application passerelle assure leur correspondance pour les fonctionnalités d'un bus de terrain/Automate programmable particulier ou directement dans un système hôte. La définition de ces applications passerelles ne relève pas du domaine d'application de la présente norme.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

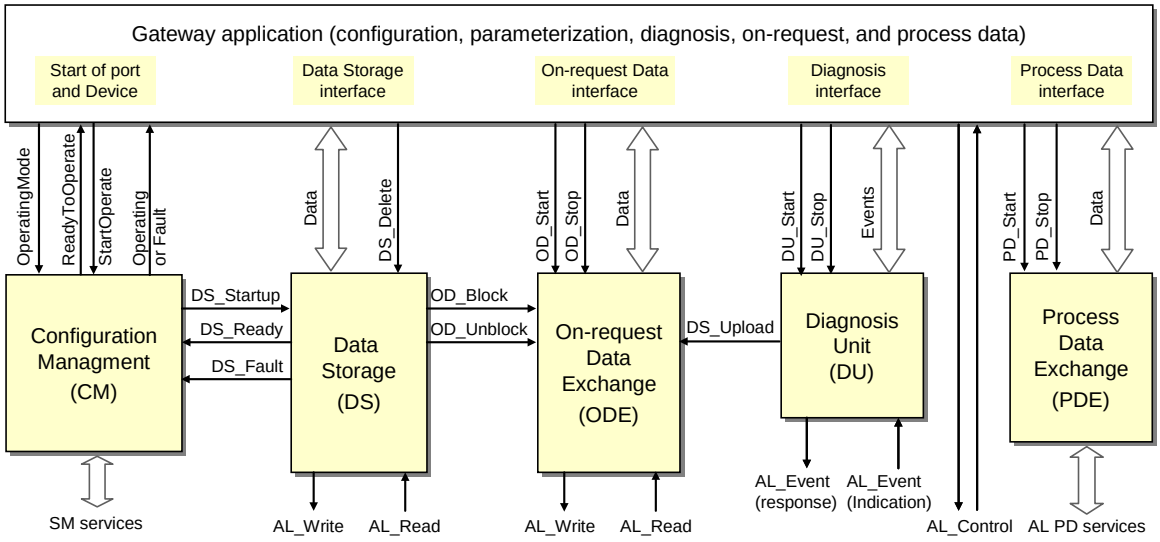
Anglais	Français
Master applications	Applications du Maître
Gateway application (configuration, parameterization, diagnosis, on-request, and process data)	Application passerelle (configuration, paramétrage, diagnostic, données à la demande et de processus)
Configuration Manager	Gestionnaire de configuration
Data Storage	Stockage des Données
On-request Exchange	Échange de données à la demande
Diagnosis Unit	Unité de Diagnostic
Process Data Exchange	Échange de Données de Processus
On-request Data objects	Objets de Données à la Demande
Process Data objects	Objets de Données de Processus
Port x Handler	Module de traitement de Port-x
Coordination	Coordination
On-request data handler	Module de traitement de Données à la Demande
Process data handler	Module de traitement des Données de Processus
Data Link Layer	Couche de liaison de données
System management	Gestion Système
Master DL-mode handler	Module de traitement de mode DL du Maître

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Message handler	Module de traitement des messages
Mode switch	Sélecteur de Mode
Inactive	Inactive
Wake-up	Réactivation
Coded switching	Commutation codée
Switching signal	Signal de commutation
Physical Layer (port)	Couche Physique (port)

Figure 94 – Structure et services d'un Maître

La Figure 95 illustre les relations entre les applications communes d'un Maître.



Légende

Anglais	Français
Gateway application (configuration, parameterization, diagnosis, on-request, and process data)	Application passerelle (configuration, paramétrage, diagnostic, données à la demande et de processus)
Start of port and Device	Démarrage du port et du Dispositif
Data Storage interface	Interface de Stockage des Données
On-request Data interface	Interface des Données à la Demande
Diagnosis interface	Interface de Diagnostic
Process Data interface	Interface des Données de Processus
Operating or Fault	En fonctionnement ou Défaut
Data	Données
Events	Événements
Configuration Management (CM)	Gestion de la Configuration (CM)
Data Storage (DS)	Stockage des Données (DS)
On-request Data Exchange (ODE)	Échange de Données à la Demande (ODE)
Diagnosis Unit (DU)	Unité de Diagnostic (DU)
Process Data Exchange (PDE)	Échange de Données de Processus (PDE)
SM services	Services SM
AL PD services	Services PD de couche AL
AL_Event (response)	AL_Event (réponse)

Figure 95 – Relations entre les applications communes d'un Maître

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Les variables internes entre les applications communes d'un Maître sont spécifiées dans le Tableau 100. La principale responsabilité est attribuée au Gestionnaire de Configuration (CM) comme illustré en Figure 95 et expliqué en 11.2.

Tableau 100 – Variables internes et Événements de contrôle des applications communes d'un Maître

Variable interne	Définition
OperatingMode	Cette variable active les ports et fournit les paramètres de configuration.
ReadyToOperate	Cette variable indique une configuration correcte du port.
StartOperate	Cette variable permet un passage explicite de tous les ports au mode OPERATE.
Operating	Cette variable indique que tous les ports sont au mode d'échange cyclique de Données de Processus
Fault	Cette variable indique l'abandon de la communication COMx à l'un des ports (voir la Figure 98 et le Tableau 101).
DS_Startup	Cette variable déclenche le diagramme d'états du Stockage des Données (DS) entraînant un téléchargement amont ou aval des paramètres du Dispositif si nécessaire (voir 11.3).
DS_Ready	Cette variable indique que le Stockage des Données a été effectué avec succès; le mode de fonctionnement est CFGCOM ou AUTOCOM (voir 9.2.2.2)
DS_Fault	Cette variable indique l'abandon du Stockage des Données suite à un défaut.
DS_Delete	Toute modification vérifiée de la configuration du Dispositif donne lieu à l'effacement du jeu de données enregistré dans le Stockage des données.
DS_Upload	Cette variable déclenche le diagramme d'états de Stockage des Données dans le Maître suite à un Événement particulier "DS_UPLOAD_REQ" provenant du Dispositif.
OD_Start	Cette variable permet d'accéder aux Données à la Demande via AL_Read et AL_Write.
OD_Stop	Cette variable indique que l'accès aux Données à la Demande via AL_Read et AL_Write a été acquitté par une réponse négative envoyée à l'application passerelle.
OD_Block	Les actions de téléchargement amont et aval des données stockées désactivent l'accès aux Données à la Demande par AL_Read ou AL_Write. L'accès par l'application passerelle est refusé.
OD_Unblock	Cette variable permet d'accéder aux Données à la Demande via AL_Read ou AL_Write.
DU_Start	Cette variable permet à l'Unité de Diagnostic de diffuser des Événements distants (Dispositif) ou locaux (Maître) vers l'application passerelle.
DU_Stop	Cette variable indique que les Événements de Dispositif ne sont pas propagés à l'application passerelle et ne sont pas acquittés. Les Événements disponibles sont bloqués jusqu'à réactivation de la DU.
PD_Start	Cette variable active l'Échange de Données de Processus avec l'application passerelle.
PD_Stop	Cette variable désactive l'Échange de Données de Processus avec l'application passerelle.

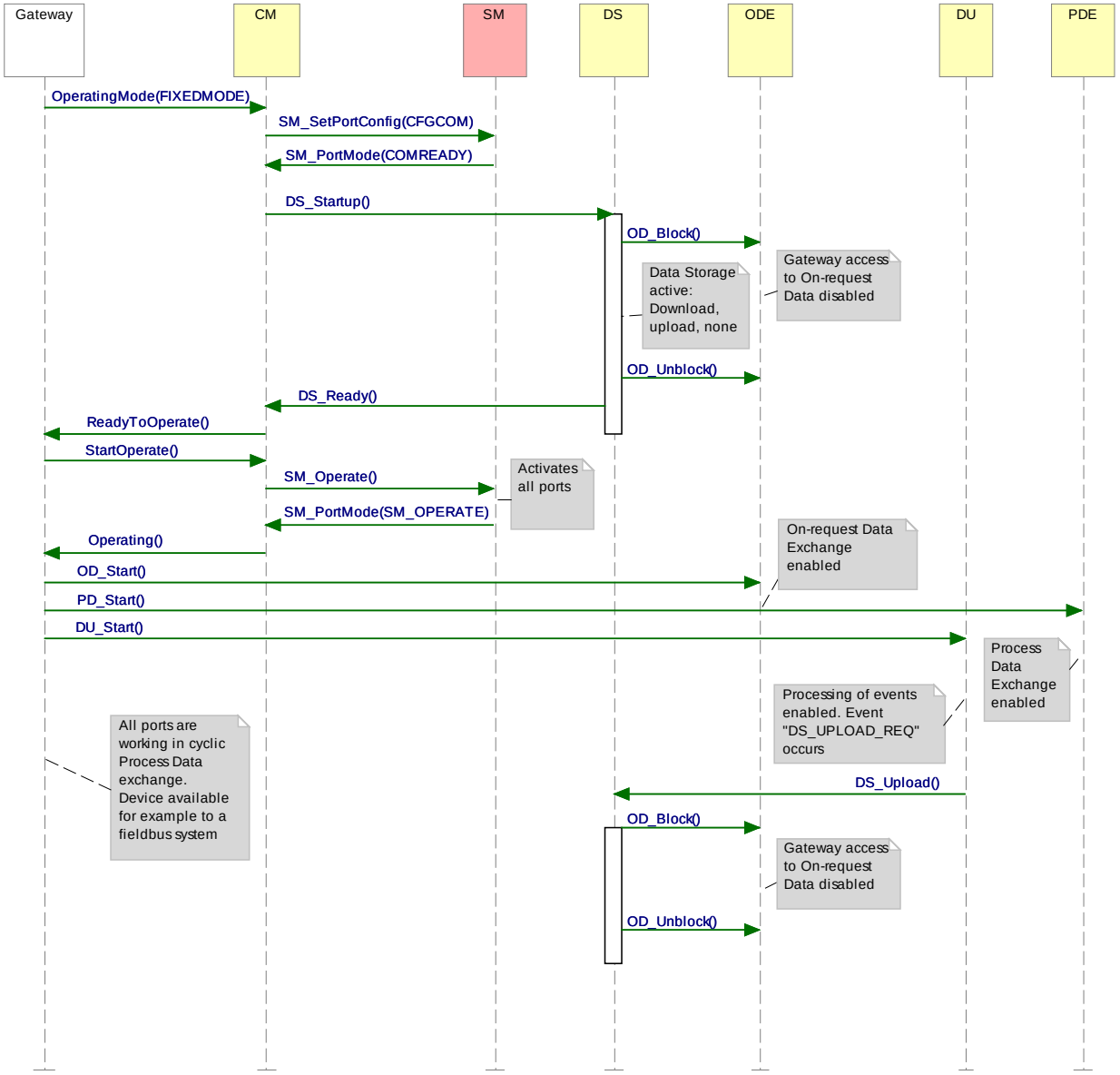
11.2 Gestionnaire de configuration (CM)

11.2.1 Généralités

La Figure 95 et la Figure 96 illustrent le rôle de coordinateur du Gestionnaire de Configuration entre toutes les applications communes de Maître. Après configuration d'un port sur les modes attribués (voir 11.2.2.1 à 11.2.2.3), le CM démarre le mécanisme de Stockage des Données (DS) et renvoie la variable "Operating" (en fonctionnement) ou "Fault" (défaut) vers l'application passerelle.

Dans le cas de la variable "Operating" d'un port particulier, l'application passerelle active les diagrammes d'états de l'Unité de Diagnostic correspondante (DU), l'Échange de Données à la Demande (ODE) et l'Échange de Données de Processus (PDE).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Gateway	Passerelle
Data Storage active:	Stockage des Données actif
Download, upload, none	Téléchargement aval, amont, aucun
Gateway access to On-request Data disabled	Accès de la passerelle aux Données à la Demande désactivé
Activates all ports	Activation de tous les ports
On-request Data Exchange enabled	Échange de Données à la Demande activé
All ports are working in cyclic Process Data exchange.	Tous les ports fonctionnent en échange cyclique de Données de Processus.
Device available for example to a fieldbus system	Dispositif disponible pour, par exemple, un système de bus de terrain
Processing of events enabled.	Traitement d'événements activé.
Event "DS_UPLOAD_REQ" occurs	L'Événement "DS_UPLOAD_REQ" a lieu
Process Data Exchange enabled	Échange de Données de Processus activé
Gateway access to On-request Data disabled	Accès de la passerelle aux Données à la Demande désactivé

Figure 96 – Diagramme de séquence des actions du Gestionnaire de Configuration

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Une fois que tous les ports de SDCI sont prêts ("ReadyToOperate", voir la Figure 96), l'application passerelle doit activer tous les ports ("StartOperate") pour s'assurer que la synchronisation des cycles de ports peut avoir lieu. Finalement, les Dispositifs échangent des Données de Processus ("Operating"). En cas de défauts, l'application passerelle reçoit "Abandon de la communication" ("INACTIVE" ou "COMLOST").

Dans le cas de SM_PortMode (COMP_FAULT, REVISION_FAULT, ou SERNUM_FAULT) conformément au 9.2.3, seul l'automate ODE doit être activé pour permettre le paramétrage.

À chaque nouveau démarrage d'un port, l'application passerelle désactivera d'abord (par exemple OD_Stop) les automates DU, ODE, et PDE correspondants.

Plusieurs paramètres sont à la disposition du Gestionnaire de Configuration pour obtenir un comportement spécifique.

11.2.2 Paramètres de Configuration

11.2.2.1 OperatingMode

L'un des modes de fonctionnement suivants peut être choisi. Tous les modes sont obligatoires.

INACTIVE
Le port SDCI est désactivé, la longueur des Données de Processus correspondantes pour l'entrée et la sortie est zéro. Le Maître ne doit avoir aucune activité sur ce port.

DO
Le port SDCI est configuré en sortie numérique (voir le Tableau 2 pour les contraintes). La longueur des Données de Processus de sortie est de 1 bit. Le Maître ne doit pas tenter de réactiver un quelconque Dispositif au niveau de ce port.

DI
Le port SDCI est configuré en entrée numérique. La longueur des Données de Processus d'entrée est de 1 bit. Le Maître ne doit pas tenter de réactiver un quelconque Dispositif au niveau de ce port.

FIXEDMODE
Un port SDCI est configuré pour une communication continue. L'identification définie est vérifiée. Une différence d'identification du Dispositif peut ou non entraîner le rejet du Dispositif, en fonction de la configuration du port (niveau de vérification, voir le Tableau 78).

SCANMODE
Le port SDCI est configuré pour une communication continue. L'identification en provenance du Dispositif est relue et peut être fournie comme la nouvelle identification définie. Dans le cas contraire, voir le Mode de Fonctionnement "FIXEDMODE".

11.2.2.2 PortCycle

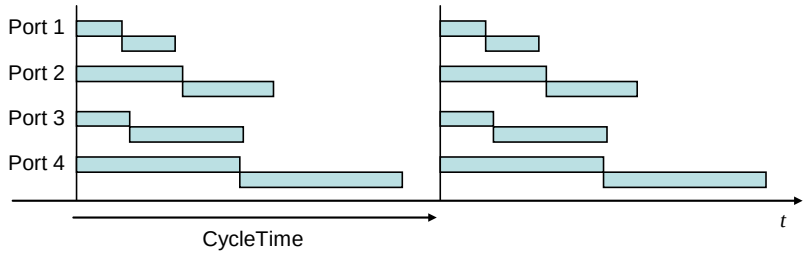
L'un des modes de cycle de port suivants peut être choisi. Aucun de ces modes n'est obligatoire.

FreeRunning
La durée de cycle du port n'est pas limitée.

FixedValue
La durée de cycle du port est fixée à une valeur spécifique. Si le Dispositif n'est pas capable de réaliser cette temporisation, par exemple si celle-ci est inférieure au MinCycleTime du Dispositif, une erreur doit être générée. La valeur fixe peut être écrite dans le paramètre CycleTime, comme spécifié en 11.2.2.3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

MessageSync
La durée de cycle du port est limitée au démarrage synchrone de tous les messages sur tous les ports SDCI de ce Maître. Dans ce cas, la durée de cycle est fournie par le MinCycleTime le plus élevé des Dispositifs connectés. Tous les ports de Maître réglés sur ce mode ont ce comportement, comme illustré en Figure 97. Les valeurs de déplacement et de gigue doivent être notées dans le manuel de l'utilisateur.



Légende

Anglais	Français
Cycle Time	Durée de cycle

Figure 97 – Ports en mode MessageSync

11.2.2.3 CycleTime

Ce paramètre contient la durée de cycle demandée ou réelle de ports spécifiques. Il doit être fourni comme une valeur ayant une résolution de 100 µs.

11.2.2.4 PDConfig

Ce jeu de paramètres contient les règles de mise en correspondance des Données de Processus entre le flux de Données de Processus du Dispositif et le flux de Données de Processus de la passerelle (pour les définitions, voir l'exemple de la Figure 107).

LenIn
Ce paramètre contient la longueur demandée, en bits, des Données de Processus d'entrée du Dispositif.

PosIn
Ce paramètre contient le décalage, en bits, dans le flux de Données de Processus d'entrée de la passerelle.

SrcOffsetIn
Ce paramètre contient le décalage, en bits, dans le flux de Données de Processus d'entrée du Dispositif.

LenOut
Ce paramètre contient la longueur demandée, en bits, des Données de Processus de sortie du Dispositif.

PosOut
Ce paramètre contient le décalage, en bits, dans le flux de Données de Processus de sortie de la passerelle.

SrcOffsetOut
Ce paramètre contient le décalage, en bits, dans le flux de Données de Processus de sortie du Dispositif.

11.2.2.5 DeviceIdentification

Ce jeu de paramètres contient l'identification configurée réelle du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

VendorID
Ce paramètre contient l'identifiant spécifique au fournisseur demandé ou lu, comme spécifié en B.1.8.

DeviceID
Ce paramètre contient l'identifiant spécifique au Dispositif demandé ou lu, comme spécifié en B.1.9.

SerialNumber

Ce paramètre contient l'identifiant spécifique au SerialNumber demandé ou lu, comme spécifié en B.2.13.

InspectionLevel

Ce paramètre contient l'InspectionLevel demandé, comme spécifié dans le Tableau 78.

11.2.2.6 DataStorageConfig

Ce jeu de paramètres contient les réglages du mécanisme de Stockage des Données (DS).

ActivationState
Ce paramètre contient l'état demandé du mécanisme DS pour ce port. Les modes suivants sont pris en charge:

DS_Enabled
Le mécanisme DS est actif et assure pleinement sa fonctionnalité, comme spécifié en 11.3.2.

DS_Disabled
Le mécanisme DS est inactif et le jeu complet de paramètres de ce port reste enregistré.

DS_Cleared
Le mécanisme DS est désactivé et le jeu de paramètres enregistré de ce port est effacé.

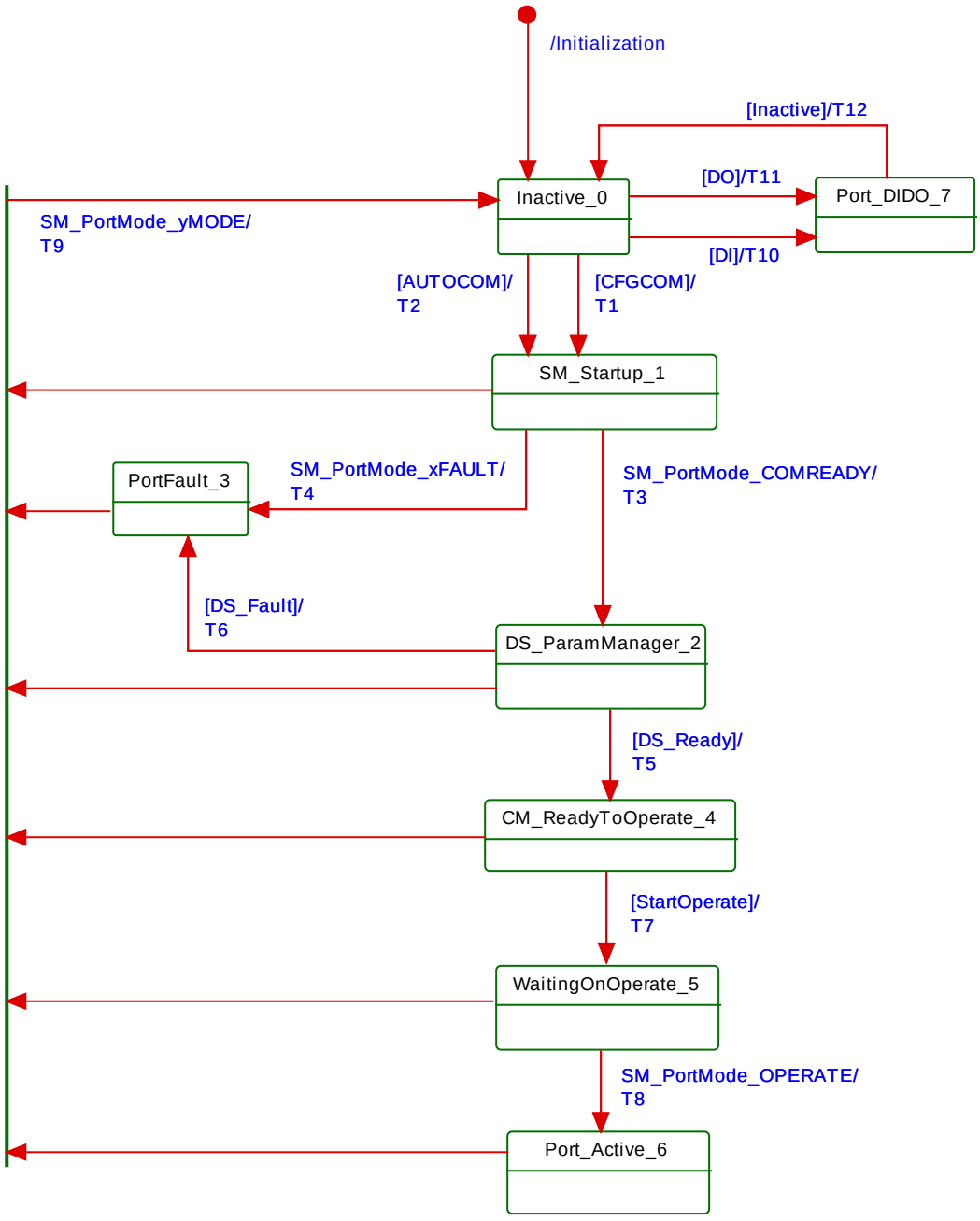
DownloadEnable
Le mécanisme DS est autorisé à écrire des données vers le Dispositif connecté.

UploadEnable
Le mécanisme DS est autorisé à lire des données à partir du Dispositif connecté.

11.2.3 Diagramme d’états du Gestionnaire de Configuration

La Figure 98 illustre le diagramme d’états du Gestionnaire de Configuration du Maître.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

xFAULT: REV_FAULT ou COMP_FAULT ou SERNUM_FAULT
yMODE: INACTIVE ou COMLOST

Anglais	Français
Initialization	Initialisation
Inactive	Inactive

Figure 98 – Diagramme d'états du Gestionnaire de Configuration

Les différents états montrent les étapes des commandes nécessaires à l'établissement ou au maintien de la communication ou de l'état DI ou DO.

Toute modification de la configuration du port peut être activée en modifiant la variable OperatingMode (voir 11.2.2.1).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau 101 illustre la table de transitions d'états du diagramme d'états du Gestionnaire de Configuration.

Tableau 101 – Tables de transitions d'états du Gestionnaire de Configuration

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente de l'une des variables OperatingMode en provenance de l'application passerelle: DO, DI, AUTOCOM, ou CFGCOM.
SM_Startup_1	Attente de l'établissement de la communication ou perte de communication ou n'importe lequel des défauts REVISION_FAULT, COMP_FAULT, ou SERNUM_FAULT (voir le Tableau 83).
DS_ParamManager_2	Attente de l'achèvement du démarrage du Stockage des Données. Les paramètres sont téléchargés dans le Dispositif (aval) ou en provenance du Dispositif (amont).
PortFault_3	Dispositif à l'état PREOPERATE (communiquant). Cependant, un des trois défauts REVISION_FAULT, COMP_FAULT, SERNUM_FAULT ou DS_Fault a eu lieu.
CM_ReadytoOperate_4	Le Port attend que l'application passerelle indique "StartOperate".
WaitingOnOperate_5	Attente du passage de la SM en mode OPERATE.
PortActive_6	Le port est en mode OPERATE. L'application passerelle échange des Données de Processus et est prête à envoyer ou recevoir des Données à la Demande.
PortDIDO_7	Le port est en mode DI ou DO. L'application passerelle échange des Données de Processus (DI ou DO).

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	SM_SetPortConfig_CFGCOM
T2	0	1	SM_SetPortConfig_AUTOCOM
T3	1	2	DS_Startup: Le diagramme d'états DS est déclenché.
T4	1	3	Indication "Fault" à l'application passerelle (REVISION_FAULT, COMP_FAULT, ou SERNUM_FAULT), voir la Figure 95.
T5	2	4	Indication destinée à l'application passerelle: ReadyToOperate
T6	2	3	Échec du Stockage des Données. Retour au jeu de paramètres précédent.
T7	4	5	SM_Operate.
T8	5	6	Indication destinée à l'application passerelle: "Operating" (voir la Figure 96).
T9	1,2,3,4,5, 6	0	SM_SetPortConfig_INACTIVE. Indication "Fault" à l'application passerelle: COMLOST ou INACTIVE
T10	0	7	SM_SetPortConfig_DI. Indication destinée à l'application passerelle: DI
T11	0	7	SM_SetPortConfig_DO. Indication destinée à l'application passerelle: DO
T12	7	0	SM_SetPortConfig_INACTIVE.

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
DS_Ready	Bool	Séquence de Stockage des Données (Téléchargement amont ou aval) exécutée. Le mode de fonctionnement du port est FIXEDMODE ou SCANMODE. Voir le Tableau 100.
DS_Fault	Bool	Voir le Tableau 100.
StartOperate	Bool	L'application passerelle a forcé le port à passer en mode OPERATE.
FIXEDMODE	Bool	Un des modes de fonctionnement (voir 11.2.2.1)
SCANMODE	Bool	Un des modes de fonctionnement (voir 11.2.2.1)
DI	Bool	Un des modes de fonctionnement (voir 11.2.2.1)
DO	Bool	Un des modes de fonctionnement (voir 11.2.2.1)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.3 Stockage des Données(DS)

11.3.1 Aperçu

Le Stockage des Données entre Maître et Dispositif est défini dans la présente norme, tandis que les mécanismes adjacents de Stockage des Données de niveau supérieur dépendent du bus de terrain ou système spécifique. Le Dispositif détient un ensemble d'objets normalisés fournissant des paramètres pour le Stockage des Données, des exigences de taille de mémoire et des informations de contrôle et d'état du mécanisme de Stockage des Données. Les différentes modifications des jeux de paramètres du Stockage des Données sont détectables via la "Somme de contrôle des paramètres" (voir 10.4.8).

11.3.2 Objet de données DS

La structure d'un objet de données "Stockage des Données" est spécifiée dans le Tableau F.1.

Le Maître doit toujours détenir les informations d'en-tête (Somme de contrôle des paramètres, VendorID et DeviceID) à des fins de vérification et de contrôle. Les informations d'objets (objets 1...n) seront stockées dans la partie mémoire non volatile du Maître (voir l'Annexe F). Avant le téléchargement aval d'un objet de données "Stockage des Données" (bloc de paramètre), le Maître vérifiera la cohérence des informations d'en-tête avec le Dispositif particulier.

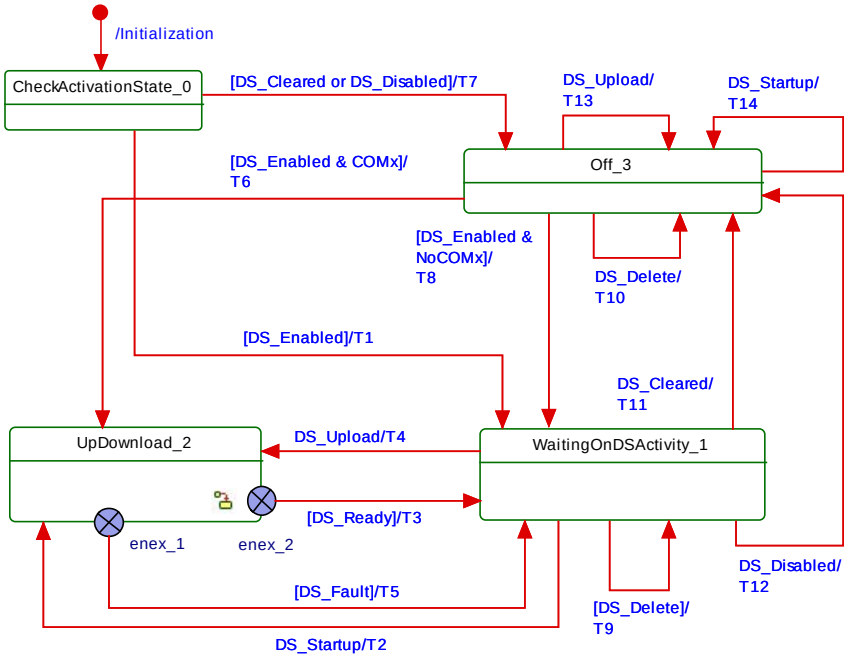
La taille maximale admise pour l'objet de données "Stockage des Données" est 2×2^{10} octets. Il est obligatoire que les Maîtres fournissent au moins cet espace mémoire pour chaque port si le mécanisme de Stockage des Données est utilisé.

11.3.3 Diagramme d'états DS

Le mécanisme de Stockage des Données est invoqué immédiatement après établissement de la communication COMx et avant passage en mode OPERATE. Pendant ce temps, toute autre communication avec le Dispositif doit être rejetée par la passerelle.

La Figure 99 illustre le diagramme d'états du mécanisme de Stockage des Données.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

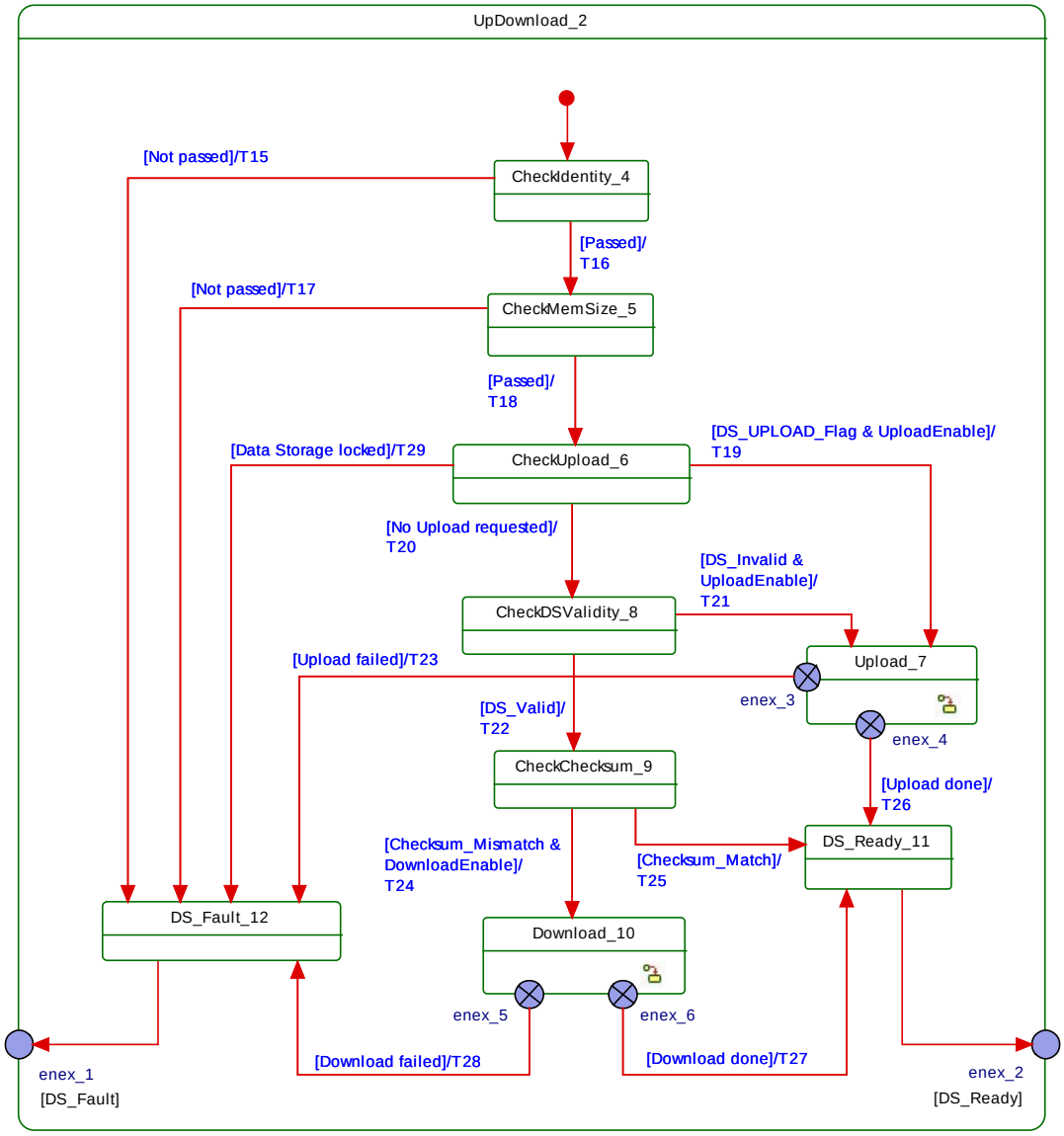
Anglais	Français
Initialization	Initialisation

Figure 99 – Principal diagramme d'états du mécanisme de Stockage des Données

La Figure 100 illustre la sous-machine de l'état "UpDownload_2".

Cette sous-machine peut être invoquée par le mécanisme de Stockage des Données ou pendant l'exécution déclenchée par un Événement "DS_UPLOAD_REQ".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

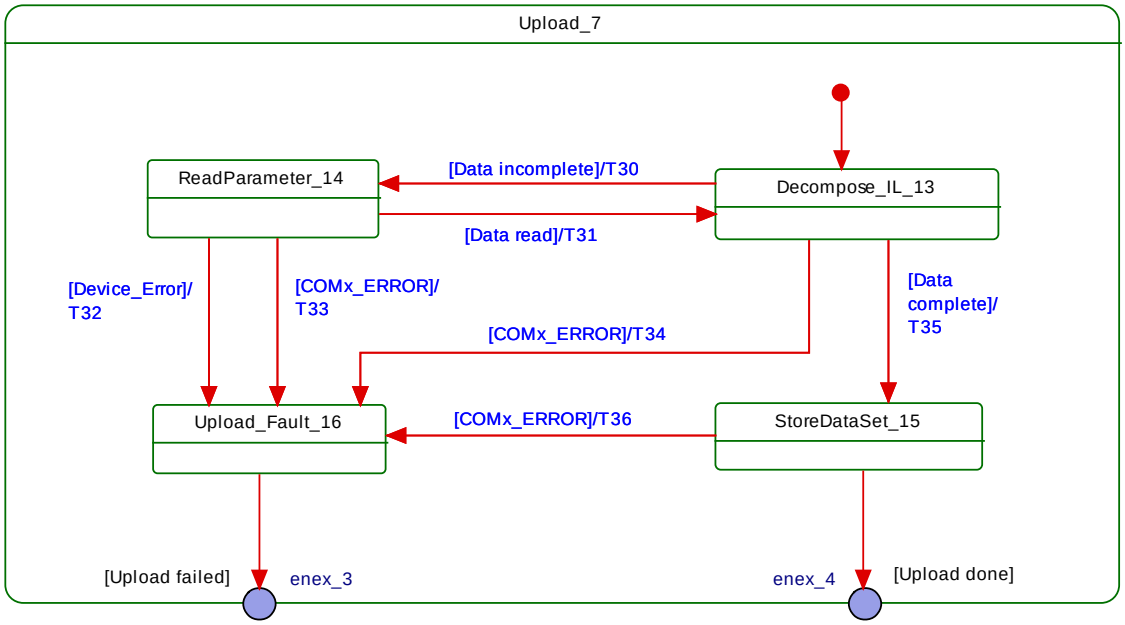
Anglais	Français
[Not passed]	[Échec]
[Passed]/	[Réussite]/
[No Upload requested]	[Pas de Téléchargement amont demandé]
[Upload failed]	[Échec du Téléchargement amont]
[Upload done]	[Téléchargement amont effectué]
[Download failed]	[Échec du Téléchargement aval]
[Download done]	[Téléchargement aval effectué]

Figure 100 – Sous-machine "UpDownload_2" du mécanisme de Stockage des Données

La Figure 101 illustre la sous-machine de l'état "Upload_7".

Ce diagramme d'états peut être invoqué par le mécanisme de Stockage des Données ou pendant l'exécution déclenchée par un Événement "DS_UPLOAD_REQ".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



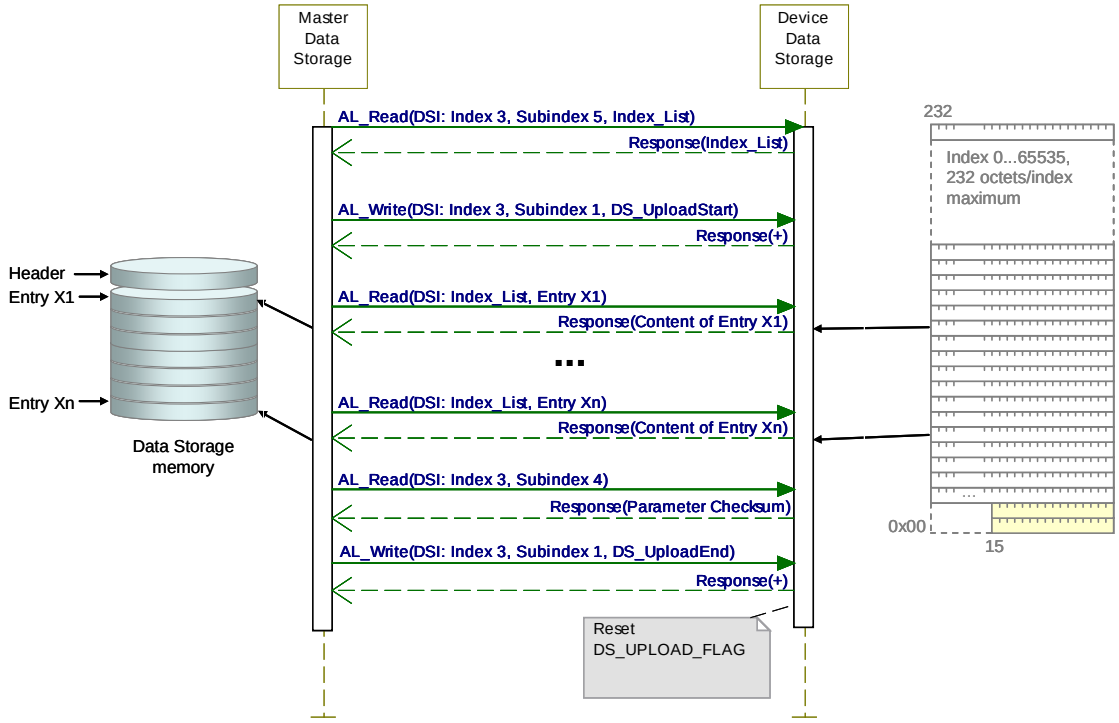
Légende

Anglais	Français
[Data incomplete]	[Données incomplètes]
[Data read]	[Lecture des Données]
[Data complete]	[Données complètes]
[Upload failed]	[Échec du Téléchargement amont]
[Upload done]	[Téléchargement amont effectué]

Figure 101 – Sous-machine du Stockage des Données"Upload_7"

La Figure 102 représente la séquence de téléchargement amont de Stockage des Données utilisant l'Index de Stockage des Données(DSI) spécifié en B.2.3 et dans le Tableau B.10. La structure de Index_List est spécifiée dans le Tableau B.11. L'indicateur DS_UPLOAD_FLAG doit être réinitialisé à la fin de chaque séquence (voir le Tableau B.10).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

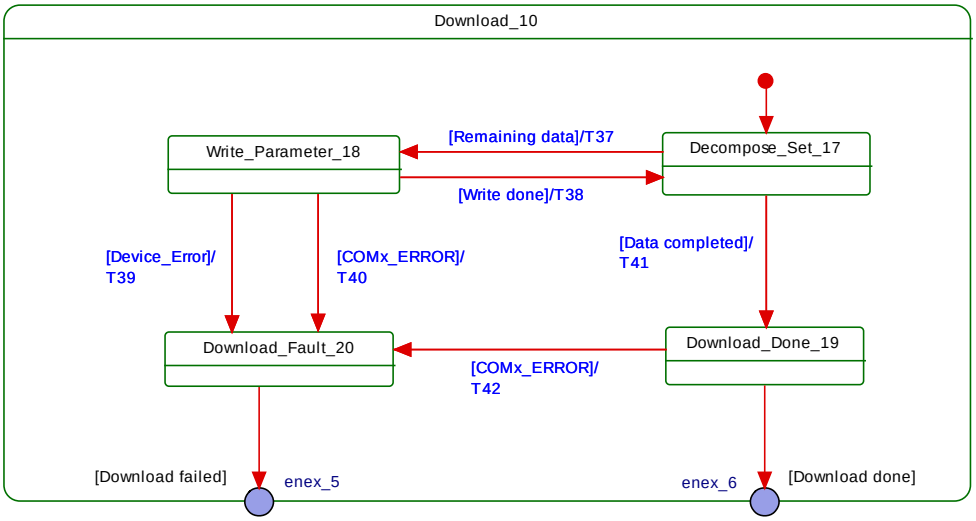
Anglais	Français
Master Data Storage	Stockage des Données du Maître
Device Data Storage	Stockage des Données du Dispositif
AL_Read(DSI: Index 3, Subindex 5, Index_List)	AL_Read(DSI: Index 3, Sous-index 5, Index_List)
Response(Index_List)	Réponse(Index_List)
AL_Write(DSI: Index 3, Subindex 1, DS_UploadStart)	AL_Write(DSI: Index 3, Sous-index 1, DS_UploadStart)
Response(+)	Réponse(+)
AL_Read(DSI: Index_List, Entry X1)	AL_Read(DSI: Index_List, Entrée X1)
Response(Content of Entry X1)	Réponse(Contenu de l'Entrée X1)
AL_Read(DSI: Index_List, Entry Xn)	AL_Read(DSI: Index_List, Entrée Xn)
Response(Content of Entry Xn)	Réponse(Contenu de l'Entrée Xn)
AL_Read(DSI: Index3, Subindex 4)	AL_Read(DSI: Index3, Sous-index 4)
Response(Parameter Checksum)	Réponse(Somme de contrôle des paramètres)
AL_Write(DSI: Index 3, Subindex 1, DS_UploadEnd)	AL_Write(DSI: Index 3, Sous-index 1, DS_UploadEnd)
Response(+)	Réponse(+)
Header	En-tête
Entry X1	Entrée X1
Entry Xn	Entrée Xn
Data Storage memory	Mémoire de Stockage des Données
Reset	Réinitialisation
DSI = Data Storage Index	DSI = Index de Stockage des Données
Index 0...65535, 232 octets/index maximum	Index 0 à 65535, 232 octets/index au maximum

Figure 102 – Diagramme de séquence du téléchargement amont de Stockage des Données

La Figure 103 illustre la sous-machine de l'état "Download_10".

Ce diagramme d'états peut être invoqué par le mécanisme de Stockage des Données.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
[Remaining Data]	[Données restantes]
[Write done]	[Écriture effectuée]
[Data completed]	[Données terminées]
[Download failed]	[Échec du Téléchargement aval]
[Download done]	[Téléchargement aval effectué]

Figure 103 – Sous-machine du Stockage des Données "Download_10"

La Figure 104 représente la séquence de téléchargement aval de Stockage des Données utilisant l'Index de Stockage des Données (DSI) spécifié en B.2.3 et dans le Tableau B.10. La structure de Index_List est spécifiée dans le Tableau B.11. L'indicateur DS_UPLOAD_FLAG doit être réinitialisé à la fin de chaque séquence (voir le Tableau B.10).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Master Data Storage	Stockage des Données du Maître
Device Data Storage	Stockage des Données du Dispositif
AL_Write(DSI: Index 3, Subindex 1, DS_ DownloadStart)	AL_Write(DSI: Index 3, Sous-index 1, DS_ DownloadStart)
Response(+)	Réponse(+)
AL_Write (DSI: Index_List, Entry X1)	AL_Write(DSI: Index_List, Entrée X1)
Response(+)	Réponse(+)
AL_Write (DSI: Index_List, Entry Xn)	AL_Write(DSI: Index_List, Entrée Xn)
AL_Read(DSI: Index3, Subindex 4)	AL_Read(DSI: Index3, Sous-index 4)
Response(Parameter Checksum)	Réponse(Somme de contrôle des paramètres)
AL_Write(DSI: Index 3, Subindex 1, DS_ DownloadEnd)	AL_Write(DSI: Index 3, Sous-index 1, DS_ DownloadEnd)
Header	En-tête
Entry X1	Entrée X1
Entry Xn	Entrée Xn
Data Storage memory	Mémoire de Stockage des Données
Index 0...65535, 232 octets/index au maximum	Index 0 à 65535, 232 octets/index au maximum

Figure 104 – Diagramme de séquence du téléchargement aval de Stockage des Données

Le Tableau 102 présente les états et transitions des diagrammes d'états de Stockage des Données.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau 102 – États et transitions des diagrammes d'états de Stockage des Données

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
CheckActivationState_0	Vérification de l'état courant de la configuration du DS: Indépendamment de l'état de la communication, DS_Startup provenant de la gestion de la configuration ou un Événement DS_UPLOAD_REQ est attendu.
WaitingOnDSActivity_1	Attente d'une demande de téléchargement amont, démarrage du Dispositif, toute modification de l'état d'activation indépendante de l'état de communication du Dispositif.
UpDownload_2	Sous-machine pour des actions de téléchargement amont/aval et des vérifications
Off_3	Traitement du Stockage des Données arrêté ou désactivé
SM: CheckIdentity_4	Vérification de l'identification du Dispositif (DeviceID, VendorID) au regard du jeu de paramètres dans le Stockage des Données (voir le Tableau F.2). L'absence de contenu ne génère pas de défaut.
SM: CheckMemSize_5	Vérification de la taille du jeu de données (Index 3, Sous-index 3) au regard de la taille de stockage disponible du Maître
SM: CheckUpload_6	Vérification de l'indicateur DS_UPLOAD_FLAG dans l'Index de Stockage des Données (voir le Tableau B.10)
SM: Upload_7	Sous-machine des actions de téléchargement amont
SM: CheckDSValidity_8	Vérification de la validité ou non-validité des données stockées dans le Maître. Un Maître pourrait être remplacé entre des activités de téléchargement amont et aval. Il incombe au concepteur d'un Maître de mettre en œuvre un mécanisme de vérification de la validité en fonction des cas d'utilisation choisis
SM: CheckChecksum_9	Vérification des différences entre le contenu du jeu de données et les paramètres du Dispositif via la "Somme de contrôle des paramètres" dans l'Index de Stockage des Données (voir le Tableau B.10)
SM: Download_10	Sous-machine des actions de téléchargement aval
SM: DS_Ready_11	Préparation de l'indication DS_Ready vers la Gestion de la Configuration (CM)
SM: DS_Fault_12	Préparation de l'indication DS_Fault à partir de "Identification_Fault", "SizeCheck_Fault", "Upload_Fault" et "Download_Fault" vers la Gestion de la Configuration (CM)
SM: Decompose_IL_13	Lecture de la Liste des Index dans l'Index de Stockage des Données (voir le Tableau B.10). Lecture du contenu, entrée par entrée, de la Liste des Index provenant du Dispositif (voir le Tableau B.11).
SM: ReadParameter_14	Attente de l'exécution du contenu lu d'une entrée de la Liste des Index provenant du Dispositif.
SM: StoreDataSet_15	Tâche de l'application passerelle: Stockage de l'ensemble du jeu de données conformément au Tableau F.1 et au Tableau F.2
SM: Upload_Fault_16	Préparation de l'indication Upload_Fault en provenance de "Device_Error" et "COM_ERROR" comme entrée pour l'indication de niveau supérieur DS_Fault.
SM: Decompose_Set_17	Écriture du jeu de données dans le Dispositif, paramètre par paramètre, conformément au Tableau F.1.
SM: Write_Parameter_18	Attente de l'achèvement de l'écriture d'un paramètre du jeu de données dans le Dispositif.
SM: Download_Done_19	Téléchargement aval achevé. Relecture de la "Somme de contrôle des paramètres" à partir de l'Index de Stockage des Données conformément au Tableau B.10. Sauvegarde de cette valeur dans le jeu de données stocké conformément au Tableau F.2.
SM: Download_Fault_20	Préparation de l'indication Download_Fault à partir de "Device_Error" et "COM_ERROR" comme entrée pour l'indication de niveau supérieur DS_Fault.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	2	-

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T3	2	1	OD_Unblock; Indication DS_Ready à la CM
T4	1	2	Confirmation de l'Événement "DS_UPLOAD_REQ"
T5	2	1	DS_Break (AL_Write, index 3, Sous-index 1); Effacement des données intermédiaires (nettoyage de la mémoire); Retour à l'état de paramètres précédent. DS_Fault (voir la Figure 95); OD_Unblock.
T6	3	2	-
T7	0	3	-
T8	3	1	-
T9	1	1	Effacement du jeu de paramètres sauvegardé (voir le Tableau F.1 et le Tableau F.2)
T10	3	3	Effacement du jeu de paramètres sauvegardé (voir le Tableau F.1 et le Tableau F.2)
T11	1	3	Effacement du jeu de paramètres sauvegardé (voir le Tableau F.1 et le Tableau F.2)
T12	1	3	-
T13	3	3	Confirmation de l'Événement "DS_UPLOAD_REQ"; pas d'autre action
T14	3	3	DS_Ready vers CM
T15	4	12	Indication DS_Fault(Identification_Fault) à l'application passerelle
T16	4	5	Lecture de la "Taille de Stockage des Données " conformément au Tableau B.10, OD_Block
T17	5	12	Indication DS_Fault(SizeCheck_Fault) à l'application passerelle
T18	5	6	Lecture de "DS_UPLOAD_FLAG" conformément au Tableau B.10
T19	6	7	Stockage des données Index 3, Sous-index 1: "DS_UploadStart" (voir le Tableau B.10)
T20	6	8	-
T21	8	7	Stockage des données Index 3, Sous-index 1: "DS_UploadStart" (voir le Tableau B.10)
T22	8	9	-
T23	7	12	Stockage des données Index 3, Sous-index 1: "DS_Break" (voir le Tableau B.10). Indication "DS_Fault(Upload)" à l'application passerelle
T24	9	10	Stockage des données Index 3, Sous-index 1: "DS_DownloadStart" (voir le Tableau B.10)
T25	9	11	-
T26	7	11	Stockage des données Index 3, Sous-index 1: "DS_UploadEnd"; Lecture de la Somme de contrôle des paramètres (voir le Tableau B.10)
T27	10	11	-
T28	10	12	Stockage des données Index 3, Sous-index 1: "DS_Break" (voir le Tableau B.10). Indication "DS_Fault(Download)" à l'application passerelle.
T29	6	12	Indication de DS_Fault (Stockage des données verrouillées)à l'application passerelle
T30	13	14	AL_Read (Liste des Index)
T31	14	13	-
T32	14	16	-
T33	14	16	-
T34	13	16	-
T35	13	15	Lecture de la "Somme de contrôle des paramètres" (voir le Tableau B.10).
T36	15	16	-
T37	17	18	Écriture du paramètre via AL_Write

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T38	18	17	-
T39	18	20	-
T40	18	20	-
T41	17	19	Stockage de données Index 3, Sous-index 1: "DS_DownloadEnd" (voir le Tableau B.10) Lecture de la "Somme de contrôle des paramètres" (voir le Tableau B.10).
T42	19	20	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
DS_Cleared	Bool	Traitement du Stockage des Données arrêté, voir 11.2.2.6
DS_Disabled	Bool	Traitement du Stockage des Données désactivé, voir 11.2.2.6
DS_Enabled	Bool	Traitement du Stockage des Données activé, voir 11.2.2.6
COMx_ERROR	Bool	Erreur détectée dans la communication COMx
Device_Error	Bool	Accès à l'Index refusé, AL_Read ou AL_Write.cnf(-) avec Code d'erreur 0x80
DS_Startup	Variable	Déclenchement à partir du diagramme d'états CM, voir la Figure 95
NoCOMx	Bool	Pas de communication COMx
COMx	Bool	La communication COMx fonctionne correctement
DS_UPLOAD_REQ	Event	Voir le Tableau D.2
UploadEnable	Bool	Configuration du traitement du Stockage des Données, voir 11.2.2.6
DownloadEnable	Bool	Configuration du traitement du Stockage des Données, voir 11.2.2.6
DS_Valid	Bool	Jeu de paramètres disponible dans le Maître valide. Voir la description de l'état "SM: CheckDSValidity_8"
DS_Invalid	Bool	Jeu de paramètres disponible dans le Maître non valide. Voir la description de l'état "SM: CheckDSValidity_8"
Checksum_Mismatch	Bool	La "Somme de contrôle des paramètres" acquise auprès du Dispositif ne correspond pas à la Somme de contrôle dans le Stockage des Données (comparaison binaire)
Checksum_Match	Bool	La "Somme de contrôle des paramètres" acquise auprès du Dispositif correspond à la Somme de contrôle dans le Stockage des Données (comparaison binaire)

11.3.4 Sélection de paramètres pour Stockage des Données

Le concepteur du Dispositif définit les paramètres qui font partie du mécanisme de Stockage des Données.

L'IODD repère tous les paramètres non inclus dans le Stockage des Données au moyen de l'attribut "excludedFromDataStorage". Cependant, le mécanisme de Stockage des Données ne doit pas tenir compte des informations de l'IODD mais plutôt de la liste de paramètres lue en provenance du Dispositif.

11.4 Échange de Données à la Demande (ODE)

La Figure 105 illustre le diagramme d'états de l'Échange de Données à la Demande du Maître. Ce comportement est obligatoire pour un Maître.

Au cours d'une transmission de données active du mécanisme de Stockage des Données, toutes les demandes de Données à la Demande sont bloquées.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

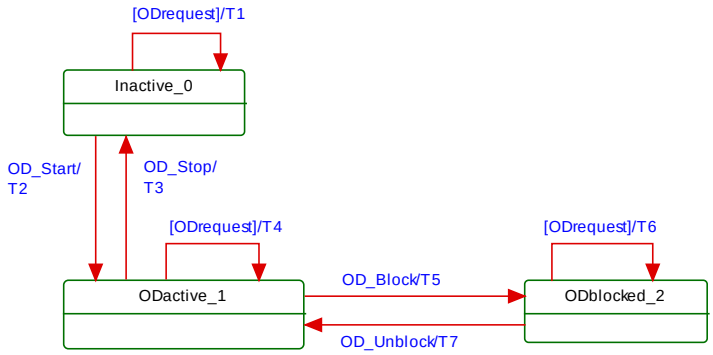


Figure 105 – Diagramme d'états de l'Échange de Données à la Demande

Le Tableau 103 présente la table de transitions d'états du diagramme d'états de l'Échange de Données à la Demande.

Tableau 103 – Table de transitions d'états du diagramme d'états de l'ODE

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Inactive_0	Attente d'activation
ODactive_1	Communication des Données à la Demande active utilisant AL_Read ou AL_Write
ODblocked_2	Communication des Données à la Demande bloquée

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	0	Accès bloqué (inactif): indique "Service non disponible" à l'application passerelle
T2	0	1	-
T3	1	0	-
T4	1	1	AL_Read ou AL_Write
T5	1	2	-
T6	2	2	Accès temporairement bloqué: indique "Service non disponible" à l'application passerelle
T7	2	1	-

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
ODrequest	Variable	Lecture ou écriture de Données à la Demande demandée via AL_Read ou AL_Write

11.5 Unité de Diagnostic (DU)

L'Unité de Diagnostic (DU) achemine des Événements de la couche AL à l'unité de Stockage des Données ou à l'application passerelle. Ces Événements contiennent principalement des informations de diagnostic.

L'objectif principal des informations de diagnostic est d'alerter efficacement un opérateur. Ce qui signifie:

- pas d'excès d'informations de diagnostic;
- compte-rendu de la cause première d'un incident dans un Dispositif ou dans le Maître et non des défauts corrélés qui en résultent;

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- Les informations de diagnostic doivent fournir des instructions quant à la manière d'assurer la maintenance ou la réparation du composant concerné pour une reprise rapide du système d'automatisation.

Dans l'interface SDCI, les informations de diagnostic des Dispositifs sont acheminées vers le Maître, via des Événements constitués de qualificatifs (EventQualifiers) et de codes (EventCodes) (voir A.6). Le texte interprétable par l'utilisateur associé à ces informations est fourni dans la présente norme pour des EventCodes normalisés (voir l'Annexe D) et dans le fichier IODD correspondant d'un Dispositif donné pour des EventCodes spécifiques au fournisseur. Les EventCodes normalisés peuvent être mis en correspondance avec des définitions de diagnostic de canal de bus de terrain sémantiquement identiques ou très proches au sein de l'application passerelle. Les codages IODD spécifiques au fournisseur peuvent être mis en correspondance avec des définitions de diagnostic de canal spécifiques (code individuel et informations correspondantes interprétables par l'utilisateur) au sein du fichier descriptif des dispositifs de bus de terrain.

Des outils techniques d'ingénierie de bus de terrain et des systèmes de surveillance des processus (interfaces homme-machine) peuvent utiliser le descriptif de dispositif de bus de terrain pour convertir les codes de diagnostic de bus de terrain reçus, en informations textuelles de diagnostic interprétables par l'utilisateur.

Un contrôle des flux d'informations de diagnostic permet d'éviter l'excès d'informations de diagnostic: en effet, un seul Événement par Dispositif peut être propagé à la fois au Maître/application passerelle.

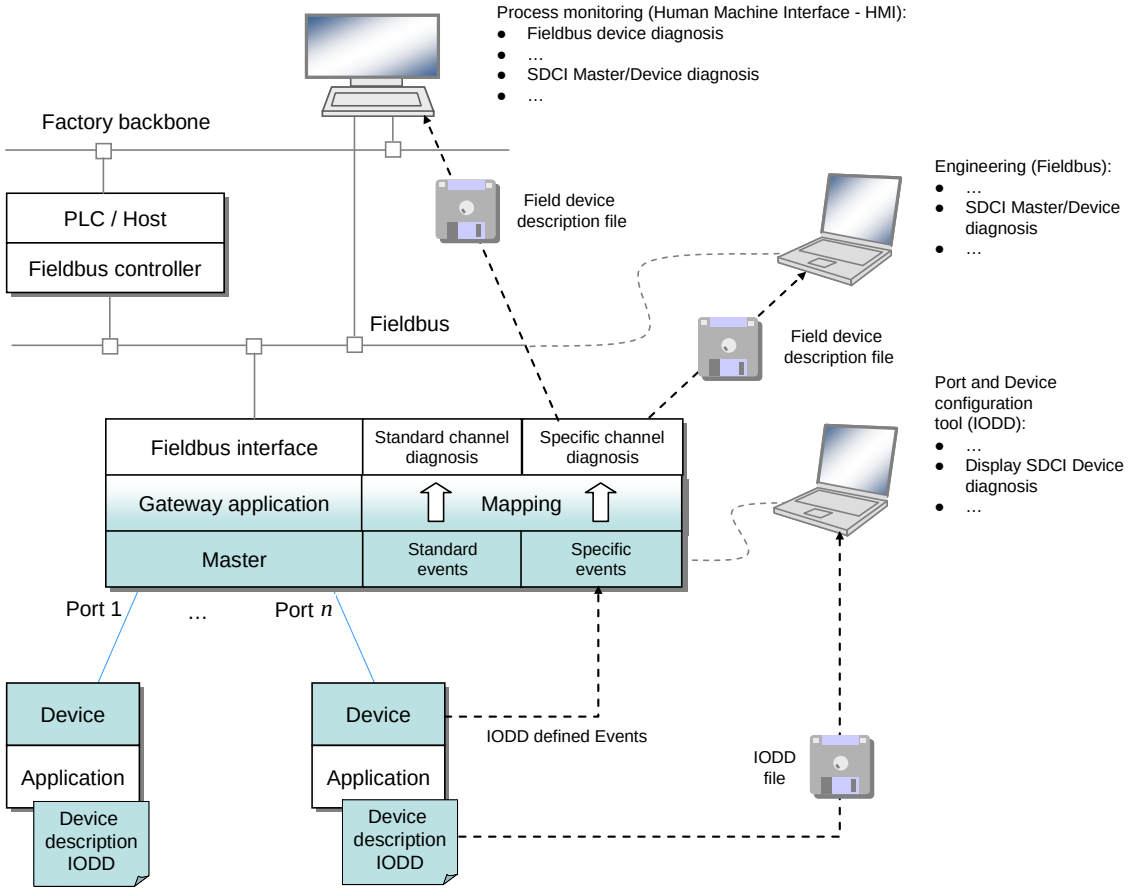
L'application passerelle est en mesure de démarrer ou d'arrêter l'Unité de Diagnostic (voir la Figure 95). Lorsque l'Unité de diagnostic est à l'arrêt, elle reporte toute indication AL_Event.ind reçue jusqu'à ce qu'elle redémarre.

L'Événement spécial DS_UPLOAD_REQ (voir 10.4 et le Tableau D.2) d'un Dispositif donné doit être redirigé vers l'application commune du Maître, à savoir le Stockage des Données. Ces Événements sont acquittés par la DU proprement dite et ne sont pas propagés à la passerelle.

La Figure 106 illustre un exemple de flux d'informations de diagnostic à travers un système SDCI/bus de terrain complet.

NOTE Le flux peut s'achever au niveau du Maître/PDCT ou être plus intégré en fonction des capacités du bus de terrain.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Process monitoring (Human Machine Interface – HMI):	Surveillance des processus (Interface homme-machine – IHM):
Fieldbus device diagnosis	Diagnostic du dispositif de bus de terrain
SDCI Master/Device diagnosis	Diagnostic Maître/Dispositif de l'interface SDCI
Factory backbone	Réseau dorsal d'usine
PLC / Host	Automate programmable / hôte
Fieldbus controller	Contrôleur de bus de terrain
Field device description file	Fichier descriptif de dispositif de terrain
Engineering (Fieldbus):	Technique (bus de terrain):
SDCI Master/Device diagnosis	Diagnostic Maître/Dispositif de l'interface SDCI
Fieldbus	Bus de terrain
Fieldbus interface	Interface de bus de terrain
Standard channel diagnosis	Diagnostic de canal normalisé
Specific channel diagnosis	Diagnostic de canal spécifique
Port and Device configuration tool (IODD):	Outil de configuration des ports et Dispositifs (IODD):
Display SDCI Device diagnosis	Diagnostic Maître/Dispositif de l'affichage
Gateway application	Application passerelle
Mapping	Mise en correspondance
Master	Maître
Standard events	Événements normalisés

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Specific events	Événements spécifiques
Device	Dispositif
Application	Application
Device description IODD	IODD de description du dispositif
IODD defined Events	Événements définis par IODD
IODD file	Fichier IODD

NOTE Les zones bleutées indiquent les fonctionnalités spécifiées dans la présente norme.

Figure 106 – Vue système de la propagation des informations de diagnostic SDCI via des Événements

11.6 Échange de PD (PDE)

11.6.1 Généralités

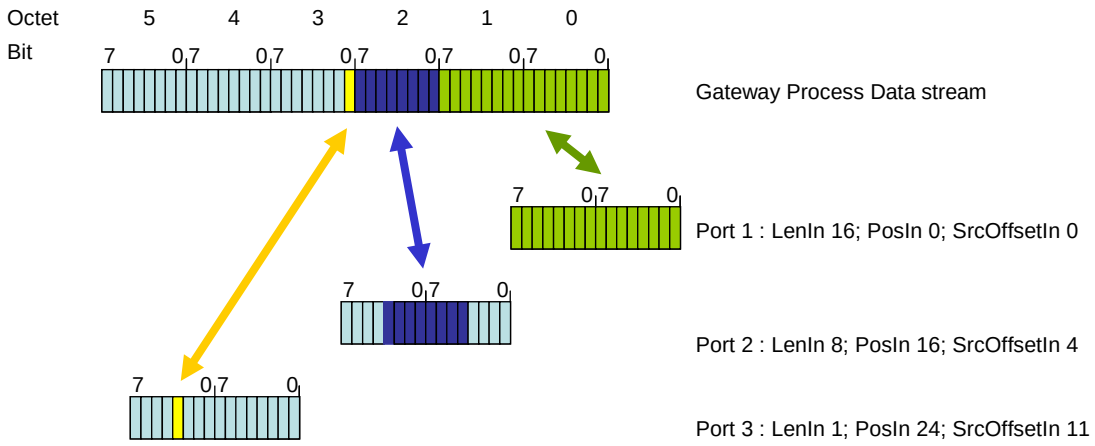
L'Échange de Données de Processus assure la transmission des Données de Processus entre l'application passerelle et le Dispositif connecté.

Après établissement de la communication et le Stockage des Données, le port est prêt pour d'éventuels transferts de Données à la Demande (ODE). La communication de Données de Processus est activée lorsque le port spécifique ou tous les ports sont en mode OPERATE.

11.6.2 Mise en correspondance des Données de Processus

Conformément à 11.2.2.4, les Données de Processus d'entrée et de sortie sont mises en correspondance avec une partie spécifique du flux de Données de Processus de la passerelle.

La Figure 107 illustre un exemple de mise en correspondance des Données de Processus en provenance de 3 ports de Maître avec le flux de Données de Processus de la passerelle.



Légende

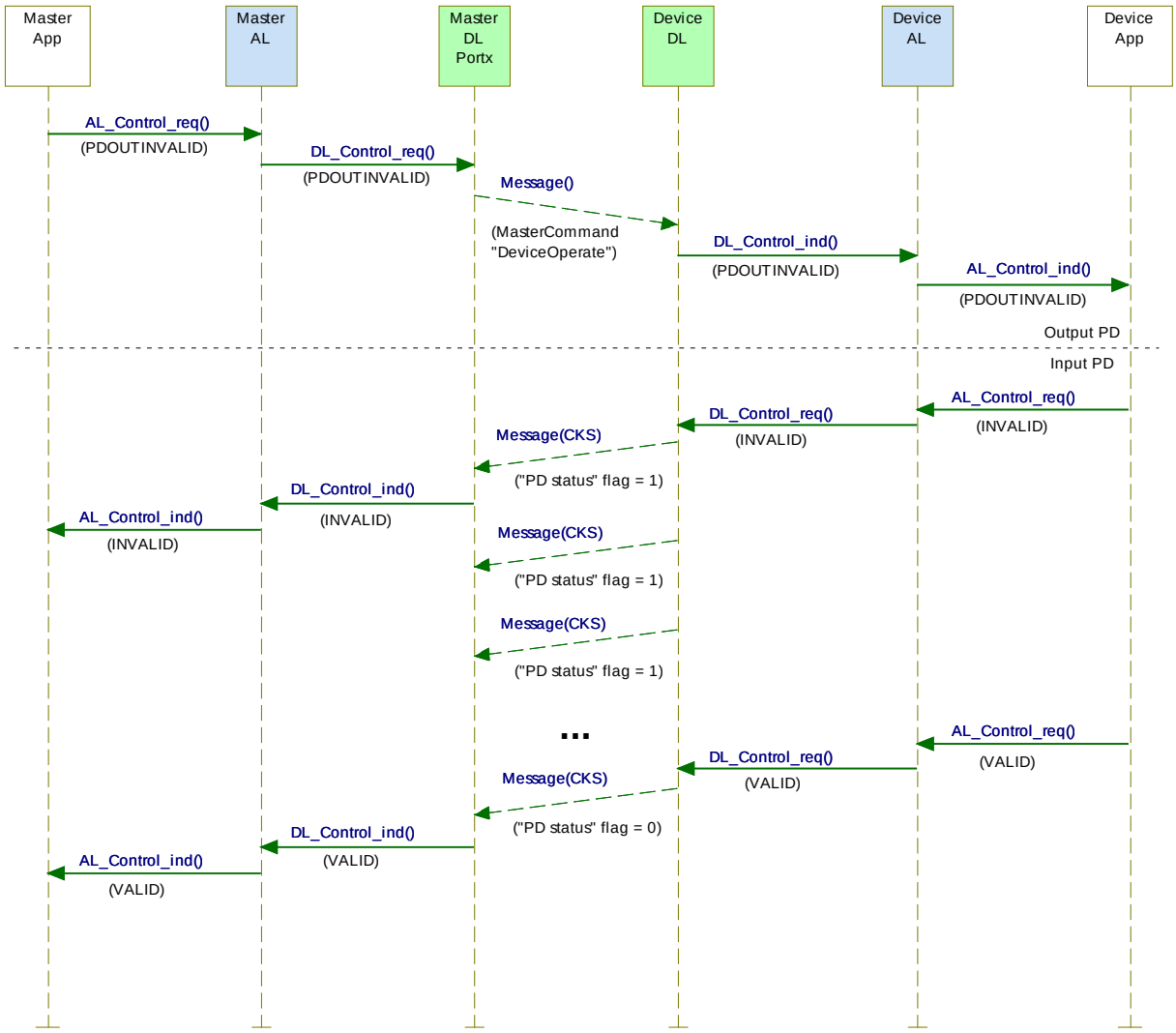
Anglais	Français
Gateway Process Data stream	Flux de Données de Processus de la passerelle

Figure 107 – Mise en correspondance des Données de Processus entre ports et flux de données de passerelle

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

11.6.3 État du qualificatif non valide/valide des Données de Processus

La partie supérieure de la Figure 108 illustre un exemple de transmission de l'état de qualificatif de PD de sortie "Non valide" envoyé de la couche AL du Maître vers la couche AL des Dispositifs.



Légende

Anglais	Français
Master App	App. du Maître
Master AL	Couche AL du Maître
Master DL Portx	Portx de la couche DL du Maître
Device DL	Couche DL du Dispositif
Device AL	Couche AL du Dispositif
Device App	Application du Dispositif
(PDOUT INVALID)	(PDOUT NON VALIDES)
(INVALID)	(NON VALIDE)
Output PD	PD de sortie
Input PD	PD d'entrée
("PD status" flag = 1)	(Drapeau "état des PD" = 1)
(VALID)	(VALIDE)
("PD status" flag = 0)	(Drapeau "état des PD" = 0)

Figure 108 – Propagation de l'état qualificatif de PD entre Maître et Dispositif

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Maître informe le Dispositif de l'état du qualificatif "valide/non valide" des Données de Processus de sortie en envoyant des MasterCommands (voir le Tableau B.2) vers la page 1 de Paramètres Directs (voir 7.3.7.1).

Pour les Données de Processus d'entrée, le Dispositif envoie l'état de qualificatif de Données de Processus dans chaque message unique comme drapeau "état des PD" dans l'octet somme de contrôle / État (CKS) octet (voir A.1.5) du message de Dispositif. La partie inférieure de la Figure 108 illustre un exemple de transmission de l'état du qualificatif de PD d'entrée "valide" envoyé de la couche AL du Dispositif vers la couche AL du Maître.

Toute perturbation en cours de transmission en mode d'entrelacement entraîne une indication de l'état qualificatif "Non valide" des Données de Processus d'entrée et de sortie respectivement.

11.7 Outil de configuration des ports et Dispositifs (PDCT)

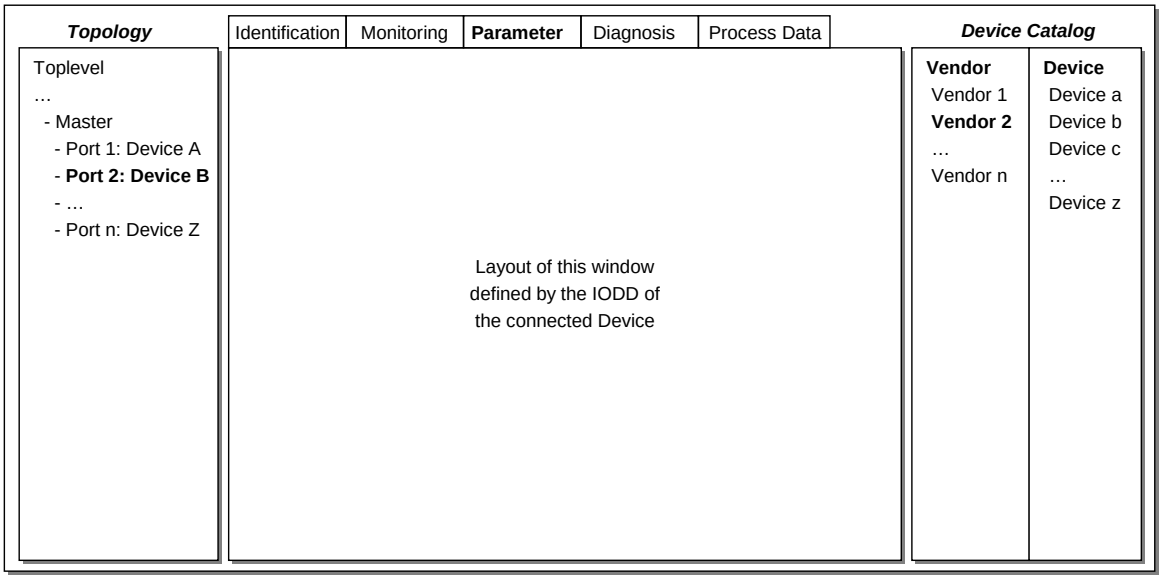
11.7.1 Généralités

La Figure 93 et la Figure 106 démontrent la nécessité d'un outil de configuration de ports, du paramétrage du Dispositif, de l'affichage des informations de diagnostic et de la fourniture des informations d'identification et de maintenance. En fonction du degré d'intégration dans un système de bus de terrain, les fonctions PDCT peuvent être réduites si, par exemple, la configuration du port peut être réalisée via le fichier descriptif de dispositif de terrain du bus de terrain particulier.

La fonctionnalité du PDCT peut être partiellement intégrée (navigation, transfert des paramètres, etc.) ou totalement intégrée dans l'outil technique du bus de terrain particulier.

11.7.2 Exemples de présentation de base

La Figure 109 illustre un exemple de présentation de l'affichage d'un PDCT.



Légende

Anglais	Français
Topology	Topologie
Identification	Identification
Monitoring	Surveillance
Parameter	Paramètre

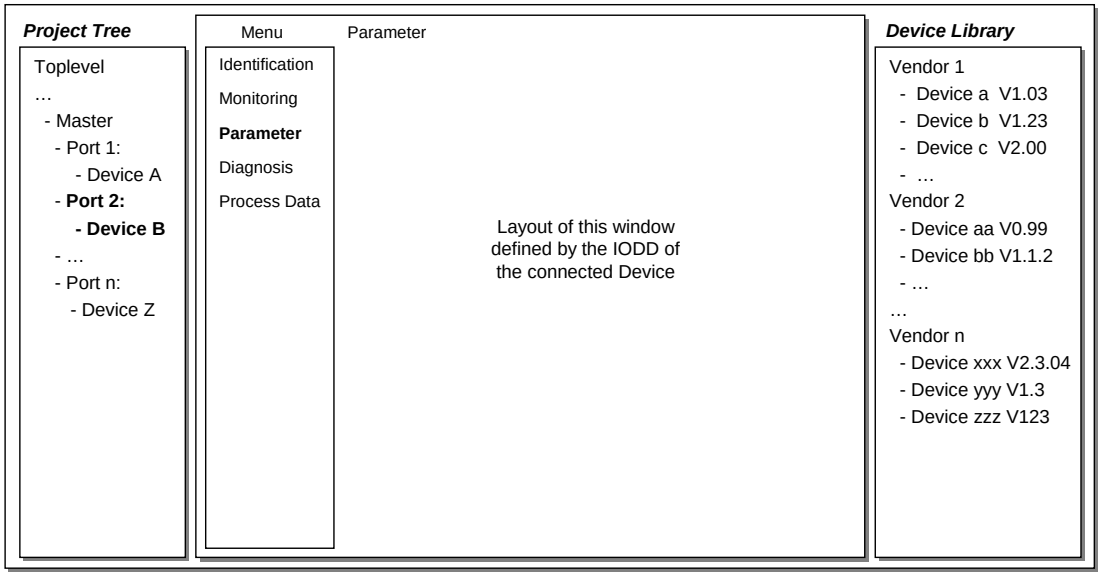
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Diagnosis	Diagnostic
Process Data	Données de processus
Device Catalog	Catalogue des dispositifs
Top level	Niveau supérieur
- Master	- Maître
- Port 1: Device A	- Port 1: Dispositif A
- Port 2: Device B	- Port 2: Dispositif B
- Port n: Device Z	- Port n: Dispositif Z
Layout of this window defined by the IODD of the connected Device	La présentation de cette fenêtre est définie par l'IODD du Dispositif connecté
Vendor	Fournisseur
Device	Dispositif

Figure 109 – Exemple 1 de présentation de l'affichage d'un PDCT

Il convient que l'affichage du PDCT dispose toujours d'une fenêtre de navigation pour une topologie de projet ou de réseau, d'une fenêtre présentant la vue particulière d'un Dispositif donné défini par son IODD et d'une fenêtre énumérant les Dispositifs disponibles sur la base des fichiers IODD installés.

La Figure 110 illustre un autre exemple de présentation de l'affichage d'un PDCT.



Légende

Anglais	Français
Project Tree	Arborescence de projet
Identification	Identification
Monitoring	Surveillance
Parameter	Paramètre
Diagnosis	Diagnostic
Process Data	Données de processus
Device library	Librairie des dispositifs
Top level	Niveau supérieur

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
- Master	- Maître
- Port 1: Device A	- Port 1: Dispositif A
- Port 2: Device B	- Port 2: Dispositif B
- Port n: Device Z	- Port n: Dispositif Z
Layout of this window defined by the IODD of the connected Device	La présentation de cette fenêtre est définie par l'IODD du Dispositif connecté
Vendor	Fournisseur
Device	Dispositif

Figure 110 – Exemple 2 de présentation de l'affichage d'un PDCT

NOTE Des informations supplémentaires sont données dans la CEI/TR 62453-61.

11.8 Application passerelle

11.8.1 Généralités

L'application passerelle dépend du système hôte particulier (bus de terrain, Automate programmable, etc.) dans lequel les applications du Maître sont intégrées. Il incombe à chaque système particulier de spécifier la mise en correspondance des services et variables du Maître.

11.8.2 Modification de la configuration du Dispositif, y compris le Stockage des Données

Après chaque modification de la configuration/paramétrage du Dispositif (CVID et/ou CDID, voir 9.2.2.2), le jeu de données correspondant précédemment enregistré dans le Maître doit être effacé ou marqué non valide via la variable DS_Delete.

11.8.3 Serveur de paramètres et contrôle des recettes

Pour ses besoins opérationnels, le Maître peut combiner l'ensemble des jeux de paramètres des Dispositifs connectés avec toutes les autres données pertinentes et les mettre à la disposition d'applications de niveau supérieur. Ces données peuvent par exemple être sauvegardées dans un serveur de paramètre auquel on peut accéder par un programme d'Automate programmable afin de modifier les paramètres de recette et ainsi assurer une fabrication polyvalente.

NOTE La structure des données échangées entre le Maître et le serveur de paramètre dépasse le cadre de la présente norme.

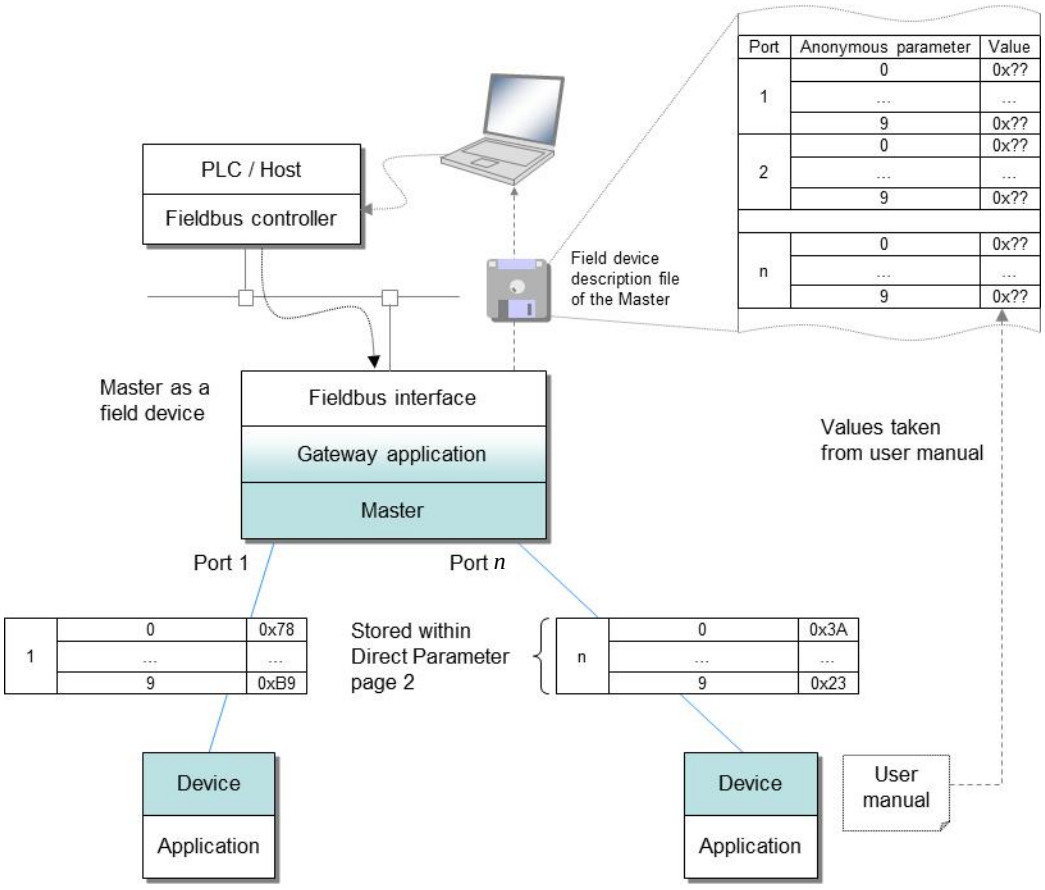
11.8.4 Paramètres anonymes

Pour certaines interfaces de passerelle, il est nécessaire de disposer d'une alternative à l'utilisation d'un outil de configuration des Ports et des Dispositifs. Pour ces interfaces, il est recommandé que l'interface de passerelle permette au système hôte de lui envoyer un bloc de 10 octets sans nom de données de configuration de Dispositif pour chaque Dispositif connecté au Maître. L'interface de passerelle utilisera alors le service AL_Write pour remettre les octets concernant chaque Dispositif à la page 2 de Paramètres Directs du Dispositif correspondant.

NOTE Les spécifications relatives à l'intégration ne sont pas couvertes par le domaine d'application de la présente norme.

Cette approche est illustrée en Figure 111.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
PLC / Host	Automate programmable / hôte
Fieldbus controller	Contrôleur de bus de terrain
Master as a field device	Maître en tant que dispositif de terrain
Fieldbus interface	Interface de bus de terrain
Gateway application	Application passerelle
Master	Maître
Stored within Direct Parameter page 2	Enregistrés dans la page 2 de Paramètres Directs
Device	Dispositif
Application	Application
Field device description file of the Master	Fichier descriptif de dispositif de terrain du Maître
User manual	Manuel de l'utilisateur
Anonymous parameter	Paramètre anonyme
Value	Valeur
Values taken from user manual	Valeurs extraites du manuel de l'utilisateur

Figure 111 – Configuration alternative de Dispositif

11.8.5 Mode port virtuel DIwithSDCI

Ce mode de fonctionnement facultatif permet d'utiliser un Dispositif ayant une capacité SIO en mode DIwithSDCI et permet au système de niveau supérieur (par exemple l'Automate programmable) d'échanger de manière acyclique des Données à la Demande. Ceci aura lieu de préférence lorsqu'il s'agit de modifier des paramètres, lors d'un arrêt de la production ou à des intervalles de diagnostic.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Ce mode de fonctionnement simplifie le programme de contrôle, car il permet d'omettre la configuration avant et après un accès acyclique.

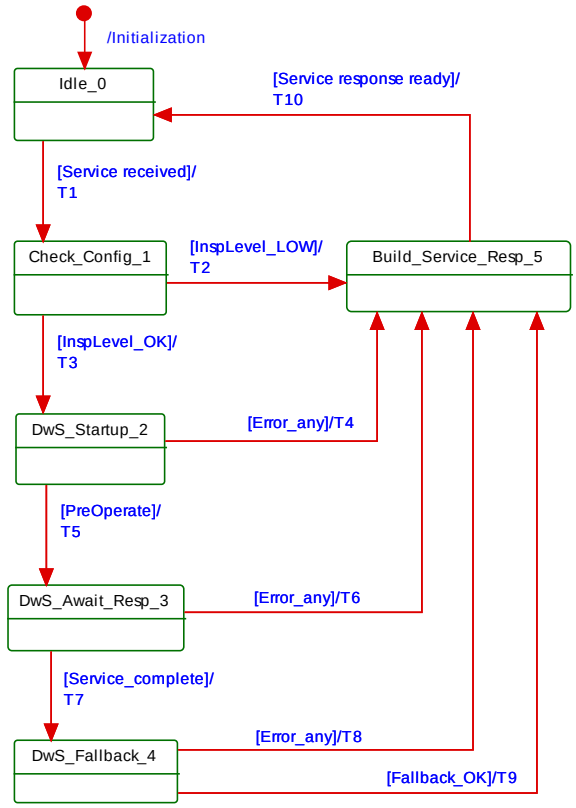
En principe, l'application passerelle exécute ce mode de fonctionnement de manière virtuelle. C'est à elle seule qu'il incombe de décider, au sein d'états particuliers, ce que peuvent être les étapes suivantes.

La CM ne connaît pas ce mode de fonctionnement. L'application passerelle lit les données de configuration détenues par la CM et utilise des services de la SM et de la couche AL pour exécuter ce mode de fonctionnement.

Les règles suivantes doivent être observées lors de la mise en œuvre du mode DIwithSDCI.

- Le signal DI du Dispositif n'est pas valide pendant l'accès acyclique de l'application passerelle.
- Il est probable qu'un signal DI non valide soit détecté très tard. Ainsi, c'est uniquement après l'accès acyclique suivant qu'un Événement "PDInvalid" peut être établi et qu'un remplacement de conducteur rompu ou de Dispositif peut être détecté.
- L'accès consommera plus de temps du fait de l'établissement de la communication et des procédures de repli, y compris le Stockage de Données.
- L'InspectionLevel doit au moins comprendre TYPE_COMP afin de détecter tout Dispositif illicite.

Le diagramme d'état présenté à la Figure 112 illustre chacun des états du mode de fonctionnement virtuel DIwithSDCI.



Légende

Anglais	Français
Initialization	Initialisation
[Service response ready]	[Réponse de service prête]
[Service received]	[Service reçu]

Figure 112 – Mode de port virtuel "DIwithSDCI"

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau 104 présente les états et transitions du mode de port virtuel "DIwithSDCI".

Tableau 104 – Transitions d'états du diagramme d'états "DIwithSDCI"

DÉNOMINATION DE L'ÉTAT	DESCRIPTION DE L'ÉTAT
Idle_0	Pour le programme de contrôle de niveau supérieur, le port est configuré en mode de fonctionnement DIwithSDCI et attend des demandes de service.
Check_Config_1	Dans cet état, le système s'assure que le niveau de vérification du port est suffisant.
DwS_StartUp_2	Dans cet état, il est effectué un démarrage complet jusqu'à l'état PREOPERATE. Ceci comprend le traitement du Stockage de données et d'Événements.
DwS_Await_Resp_3	Attente de réponses (AL-Read.rsp ou AL-Write.rsp).
DwS_FallBack_4	Après réalisation des services, le Dispositif rebascule en mode SIO via la commande MasterCommand "Repli" et le port sera configuré en DI.
Build_Service_Resp_5	La réponse de service (positive ou négative) sera générée pour finaliser le service destiné au programme de contrôle de niveau supérieur.

TRANSITION	ÉTAT DE LA SOURCE	ÉTAT DE LA CIBLE	ACTION
T1	0	1	-
T2	1	5	-
T3	1	2	Invocation de SM_SetPortConfig (en COMx)
T4	2	5	Invocation de SM_SetPortConfig (en DI)
T5	2	3	Invocation de Service (AL_Read.req ou AL_Write.req)
T6	3	5	Invocation de SM_SetPortConfig (en DI)
T7	3	4	-
T8	4	5	Invocation de SM_SetPortConfig (en DI)
T9	4	5	Invocation de SM_SetPortConfig (en DI)
T10	5	0	Invocation de la réponse de service AL correspondante

ÉLÉMENTS INTERNES	TYPE	DÉFINITION
InspLevel_LOW	Bool	Le niveau de vérification nécessaire n'est pas configuré pour détecter le Dispositif correct.
InspLevel_OK	Bool	Le niveau de vérification nécessaire est configuré pour détecter le Dispositif correct.
PreOperate	Bool	L'état PREOPERATE est établi
Service_complete	Bool	L'application passerelle a reçu AL_Read.rsp ou AL_Write.rsp
Fallback_OK	Bool	Le repli s'est achevé avec succès
Error_any	Bool	Toute erreur mettra fin à cet état

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe A
(normative)

Codages, contraintes temporelles et erreurs

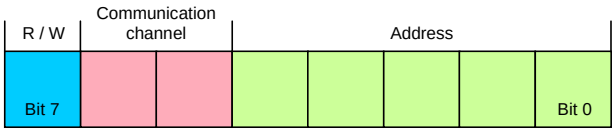
A.1 Structure générale et codage de séquences M

A.1.1 Aperçu

Le concept général des séquences M est présenté en 7.3.3.2. Les paragraphes A.1.2 à A.1.6 fournissent une description détaillée des différents éléments de séquences M.

A.1.2 Contrôle de séquence M (MC)

Le Maître indique la manière dont les données utilisateur (voir A.1.4) doivent être transmises dans un octet de contrôle de séquence M. Cette indication inclut le sens de transmission (lecture ou écriture), le canal de communication et l'adresse (décalage) des données sur le canal de communication. La structure de l'octet de contrôle de séquence M est illustrée en Figure A.1.



Légende

Anglais	Français
Communication channel	Canal de communication
Address	Adresse

Figure A.1 – Contrôle de séquence M

Bits 0 à 4: Adresse

Ces bits indiquent l'adresse, c'est-à-dire le décalage d'octet des données utilisateur sur le canal de communication spécifié (voir également le Tableau A.1). Dans le cas d'un canal ISDU, ces bits sont utilisés pour le contrôle de flux des données ISDU. L'adresse, en l'occurrence la position des données utilisateur au sein de l'ISDU, n'est disponible qu'indirectement (voir 7.3.6.2).

Bits 5 à 6: Canal de communication

Ces bits indiquent le canal de communication pour l'accès aux données utilisateur. Les valeurs définies pour le paramètre de canal de communication sont énumérées dans le Tableau A.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau A.1 – Valeurs du canal de communication

Valeur	Définition
0	Processus
1	Page
2	Diagnostic
3	ISDU

Bit 7: R/W (Lecture/Écriture)

Ce bit indique le sens de transmission des données utilisateur sur le canal de communication sélectionné, c'est-à-dire accès en lecture (transmission des données utilisateur du Dispositif vers le Maître) ou accès en écriture (transmission des données utilisateur du Maître vers le Dispositif). Les valeurs définies pour le paramètre R/W (Lecture/Écriture) sont énumérées dans le Tableau A.2.

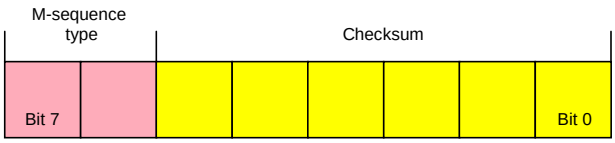
Tableau A.2 – Valeurs de R/W

Valeur	Définition
0	Accès en écriture
1	Accès en lecture

Il n'est pas exigé qu'un Dispositif prenne en charge chacune des 256 valeurs de l'octet de contrôle de séquence M. Pour l'accès en lecture vers des adresses ou des canaux de communication non utilisés, la valeur "0" doit être renvoyée. Un accès en écriture vers des adresses ou des canaux de communication non utilisés doit être ignoré.

A.1.3 Somme de contrôle / Type de séquence M (CKT)

Le type de séquence M est transmis avec la somme de contrôle dans l'octet contrôle/type. La structure de cet octet est présentée en Figure A.2.



Légende

Anglais	Français
M-sequence type	Type de séquence M
Checksum	Somme de contrôle

Figure A.2 – Octet de Somme de contrôle/Type de séquence M

Bits 0 à 5: Somme de contrôle

Ces bits contiennent une somme de contrôle de message sur 6 bits pour garantir l'intégrité des données; voir également A.1.6 et H.1.

Bits 6 à 7: Type de séquence M

Ces bits indiquent le type de séquence M. Ici c'est le Maître qui spécifie la manière dont les messages sont structurés dans la séquence M. Les valeurs définies pour le paramètre de type de séquence M sont énumérées dans le Tableau A.3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau A.3 – Valeurs de types de séquence M

Valeur	Définition
0	Type 0
1	Type 1
2	Type 2 (voir la NOTE)
3	Réservé
NOTE Les sous-types dépendent de la configuration et du sens des PD.	

A.1.4 Données utilisateur (PD ou OD)

L'expression "données utilisateur" est un terme générique utilisé pour désigner à la fois les Données de Processus et les Données à la Demande. La longueur des données utilisateur peut varier de 0 à 64 en fonction du type de séquence M et du sens de transmission (lecture/écriture). Un aperçu des types de données disponibles est fourni dans le Tableau A.4. Ces types de données peuvent être présentés comme des enregistrements (types différents) ou en matrices (types identiques).

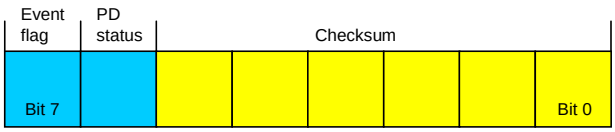
Tableau A.4 – Types de données pour les données utilisateur

Type de données	Référence
BooleanT	Voir E.2.2
UIntegerT	Voir E.2.3
IntegerT	Voir E.2.4
StringT	Voir E.2.6
OctetStringT	Voir E.2.7
Float32T	Voir E.2.5
TimeT	Voir E.2.8
TimeSpanT	Voir E.2.9

Le codage détaillé des types de données est fourni en Annexe E.

A.1.5 Somme de contrôle / état (CKS)

L'octet de somme de contrôle/état fait partie du message de réponse du Dispositif au Maître. Sa structure est illustrée en Figure A.3. Il est constitué d'une somme de contrôle sur 6 bits, d'un drapeau indiquant la validité ou l'invalidité des Données de Processus et d'un drapeau d'Événement.



Légende

Anglais	Français
Event flag	Drapeau d'Événement
PD status	État des PD
Checksum	Somme de contrôle

Figure A.3 – Octet de Somme de contrôle/État

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Bits 0 à 5: Somme de contrôle

Ces bits contiennent une somme de contrôle sur 6 bits pour garantir l'intégrité des données du message de réponse. Voir également A.1.6 et H.1.

Bit 6: État des PD

Ce bit indique si le Dispositif peut ou non fournir des Données de Processus valides. Les valeurs définies pour ce paramètre sont énumérées dans le Tableau A.5.

Ce drapeau d'état des PD doit être utilisé pour des Dispositifs ayant des Données de Processus d'entrée. Les Dispositifs ayant des Données de Processus de sortie doivent toujours indiquer "Données de Processus valides".

Si le drapeau d'état des PD est mis à “Données de Processus non valides” dans un message, toutes les Données de Processus d'entrée de l'ensemble du cycle de Données de Processus sont non valides.

Tableau A.5 – Valeurs d'état des PD

Valeur	Définition
0	Données de Processus valides
1	Données de Processus non valides

Bit 7: Drapeau d’Événement

Ce bit indique une initiative du Dispositif pour la catégorie de données "Événement" à récupérer par le Maître via le canal de communication de diagnostic (voir le Tableau A.1). Le Dispositif peut rendre compte des informations de diagnostic telles que les erreurs, les avertissements ou les notifications via des messages de réponse d'Événements. Les valeurs admissibles pour ce paramètre sont énumérées dans le Tableau A.6.

Tableau A.6 – Valeurs du drapeau d’Événement

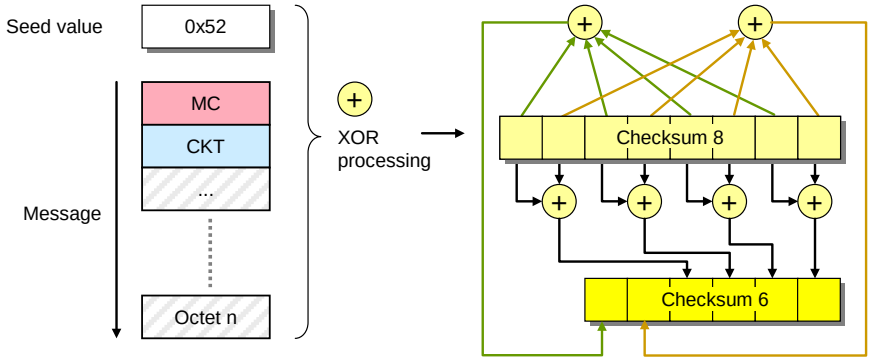
Valeur	Définition
0	Pas d'Événement
1	Événement

A.1.6 Calcul de la somme de contrôle

Le message somme de contrôle fournit une protection d'intégrité des données pour la transmission des données du Maître vers le Dispositif et du Dispositif vers le Maître. Chaque octet de données UART est protégé par le bit de parité de l'UART (voir la Figure 18). Outre cette protection de chaque octet de données, tous les octets de données de l'UART dans un message sont traités dans un message XOR (ou exclusif) octet par octet. Cet octet de contrôle/type est inclus avec les bits de somme de contrôle mis à "0". L'octet de somme de contrôle qui en résulte est compressé de 8 à 6 bits conformément à la procédure de conversion de la Figure A.4 et à ses formules associées (voir les équations en (A.1)). La somme de contrôle compressée sur 6 bits "Checksum6" est placée dans l'octet somme de contrôle/type de séquence M (voir la Figure A.2). La même procédure est utilisée pour protéger le message du Dispositif au Maître. Dans ce cas, la somme de contrôle compressée est placée dans l'octet de somme de contrôle/état (voir la Figure A.3).

Une valeur de départ de 0x52 est utilisée pour le calcul de la somme de contrôle sur l'ensemble du message. Elle est traitée en XOR avec le premier octet du message (FC).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Seed value	Valeur de départ
XOR processing	Traitement XOR
Checksum	Somme de contrôle

Figure A.4 – Principe de calcul et de compression de la somme de contrôle

Le jeu d'équations donné en (A.1) définit de manière détaillée la procédure de compression de 8 en 6 bits.

$$
\begin{aligned}
D5_6 &= D7_8 \text{ xor } D5_8 \text{ xor } D3_8 \text{ xor } D1_8 \\
D4_6 &= D6_8 \text{ xor } D4_8 \text{ xor } D2_8 \text{ xor } D0_8 \\
D3_6 &= D7_8 \text{ xor } D6_8 \\
D2_6 &= D5_8 \text{ xor } D4_8 \\
D1_6 &= D3_8 \text{ xor } D2_8 \\
D0_6 &= D1_8 \text{ xor } D0_8
\end{aligned}
\qquad \text{(A.1)}
$$

A.2 Types de séquence M

A.2.1 Aperçu

Les Données de Processus et les Données à la Demande utilisent des canaux de communication cyclique et acyclique séparés (voir la Figure 7) pour assurer une remise planifiée et déterministe des Données de Processus, sachant que la remise des Données à la Demande n'a pas d'effet sur les performances de transmission des Données de Processus.

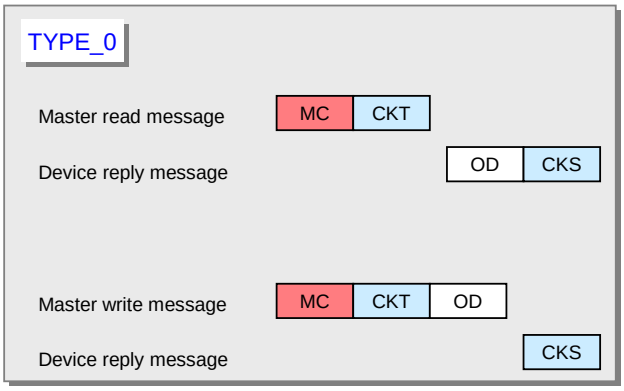
Dans l'interface SDCI, des séquences M donnent accès aux canaux de communication via l'octet de Contrôle de séquence M. Le nombre des différents types de séquences M répondent aux diverses exigences des capteurs et actionneurs en termes de taille de Données de Processus. La Figure 37 donne un aperçu des types de séquences M disponibles définis de A.2.2 à A.2.5. Le paragraphe A.2.6 fournit les règles d'utilisation des types de séquences M.

A.2.2 Séquence M de TYPE _0

La séquence M de TYPE _0 est obligatoire pour tous les Dispositifs.

La séquence M de TYPE_0 transmet uniquement des Données à la Demande. Un octet de données utilisateur est lu ou écrit par cycle. Cette séquence M est illustrée en Figure A.5.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

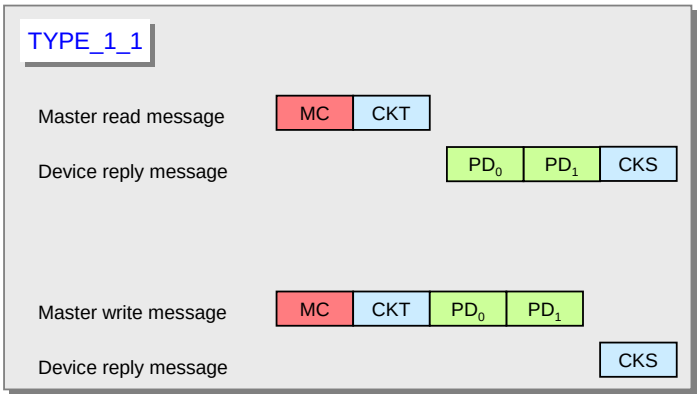
Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.5 – Séquence M de TYPE_0

A.2.3 Séquence M de TYPE_1_x

La séquence M de TYPE_1_x est facultative pour tous les Dispositifs.

La séquence M de TYPE_1_1 est illustrée en Figure A.6.



Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.6 – Séquence M de TYPE_1_1

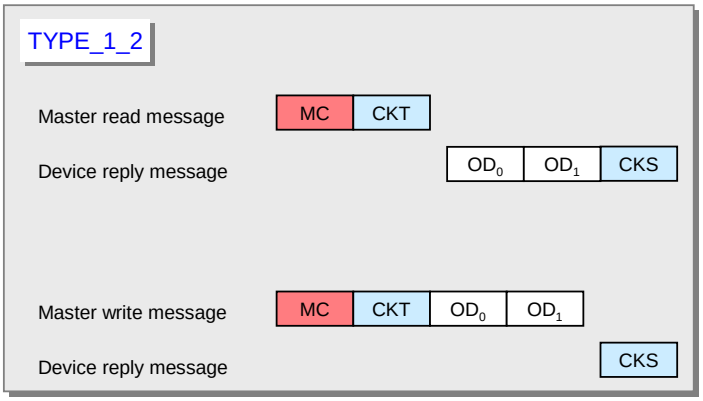
Deux octets de Données de Processus sont lus ou écrits par cycle. L'adresse (décalage binaire) appartient au canal de communication de processus (voir A.2.1).

En mode d'entrelacement (voir 7.3.4.2) et lorsque la longueur de PD est impaire, les octets restants dans les messages sont remplis de 0x00.

La séquence M de TYPE_1_2 est illustrée en Figure A.7. Deux octets de Données à la Demande sont lus ou écrits par cycle.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Pour l'accès en écriture des Données à la Demande via les canaux de communication de page et de diagnostic, seul est évalué le premier octet des Données à la Demande. Le Dispositif doit ignorer les octets restants. Le Maître doit envoyer tous les autres OD avec "0x00".



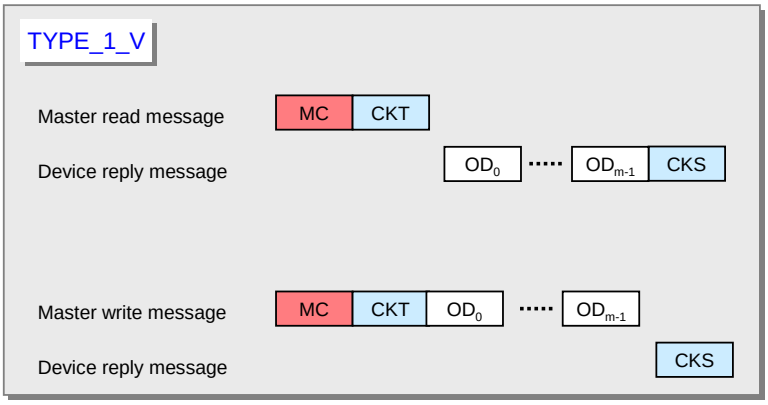
Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.7 – Séquence M de TYPE_1_2

La séquence M de TYPE_1_V qui fournit une longueur de message variable (extensible) est illustrée en Figure A.8. Un certain nombre d'octets m de Données à la Demande est lu ou écrit par cycle.

Pour l'accès en écriture des Données à la Demande via les canaux de communication de page et de diagnostic, seul est évalué le premier octet (OD_0) des Données à la Demande. Le Dispositif doit ignorer les octets restants. Le Maître doit envoyer tous les autres OD avec "0x00".



Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

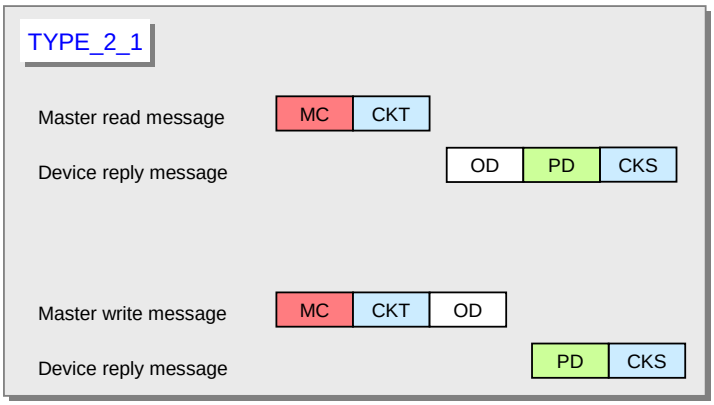
Figure A.8 – Séquence M de TYPE_1_V

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.2.4 Séquence M de TYPE_2_x

La séquence M de TYPE_2_x est facultative pour tous les Dispositifs. Les séquences M de TYPE_2_1 à TYPE_2_6 sont définies. La séquence M de TYPE_2_V fournit une longueur de message variable (extensible). La séquence M de TYPE_2_x transmet des Données de Processus et des Données à la Demande dans un message. Le nombre de Données de Processus et de Données à la Demande lu ou écrit au cours de chaque cycle dépend du type. Le paramètre d'adresse (voir la Figure A.1) appartient, dans ce cas, au canal de communication des Données à la Demande. Il est spécifié que l'adresse des Données de Processus commence implicitement à "0". Le format des Données de Processus caractérise la séquence M de TYPE_2_x.

La séquence M de TYPE_2_1 transmet, par cycle, un octet de Données de Processus de lecture et un octet de Données à la Demande de lecture ou d'écriture. Ce type de séquence M est illustré en Figure A.9.



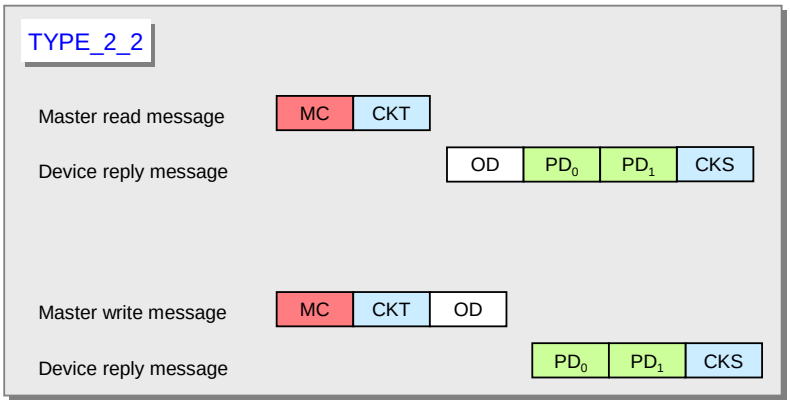
Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.9 – Séquence M de TYPE_2_1

La séquence M de TYPE_2_2 transmet, par cycle, 2 octets de Données de Processus de lecture et un octet de Données à la Demande. Ce type de séquence M est illustré en Figure A.10.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

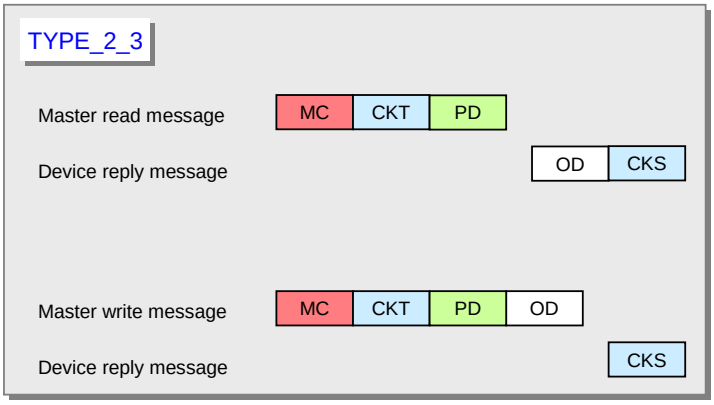


Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.10 – Séquence M de TYPE_2_2

La séquence M de TYPE_2_3 transmet, par cycle, un octet de Données de Processus d'écriture et un octet de Données à la Demande de lecture ou d'écriture. Ce type de séquence M est illustré en Figure A.11.



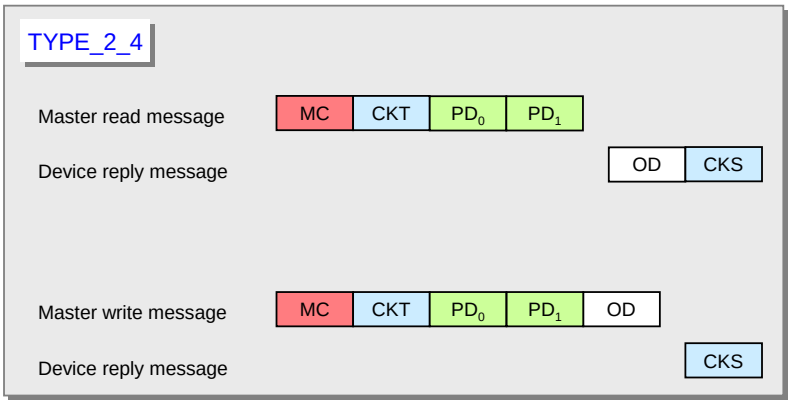
Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.11 – Séquence M de TYPE_2_3

La séquence M de TYPE_2_4 transmet, par cycle, 2 octets de Données de Processus d'écriture et un octet de Données à la Demande de lecture ou d'écriture. Ce type de séquence M est illustré en Figure A.12.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

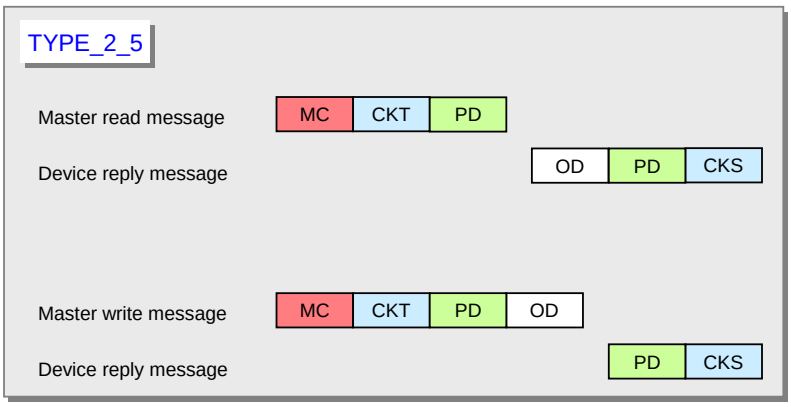


Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.12 – Séquence M de TYPE_2_4

La séquence M de TYPE_2_5 transmet, par cycle, un octet de Données de Processus d'écriture et de lecture, et un octet de Données à la Demande de lecture ou d'écriture. Ce type de séquence M est illustré en Figure A.13.



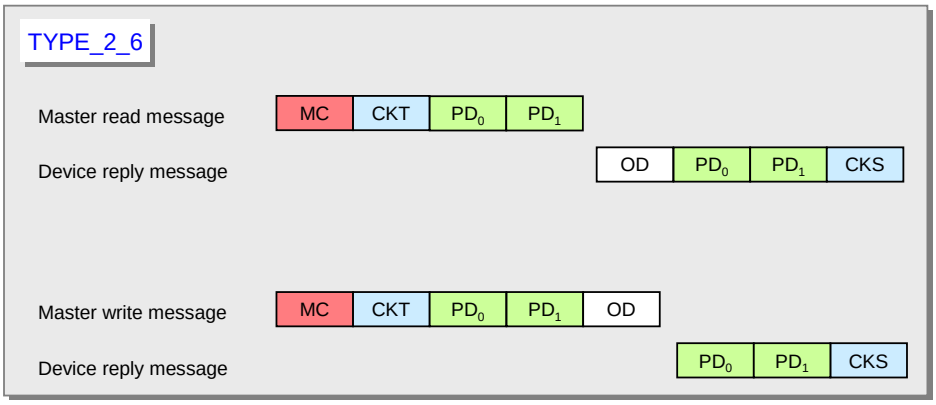
Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.13 – Séquence M de TYPE_2_5

La séquence M de TYPE_2_6 transmet, par cycle, 2 octets de Données de Processus d'écriture et de lecture, et un octet de Données à la Demande de lecture ou d'écriture. Ce type de séquence M est illustré en Figure A.14.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

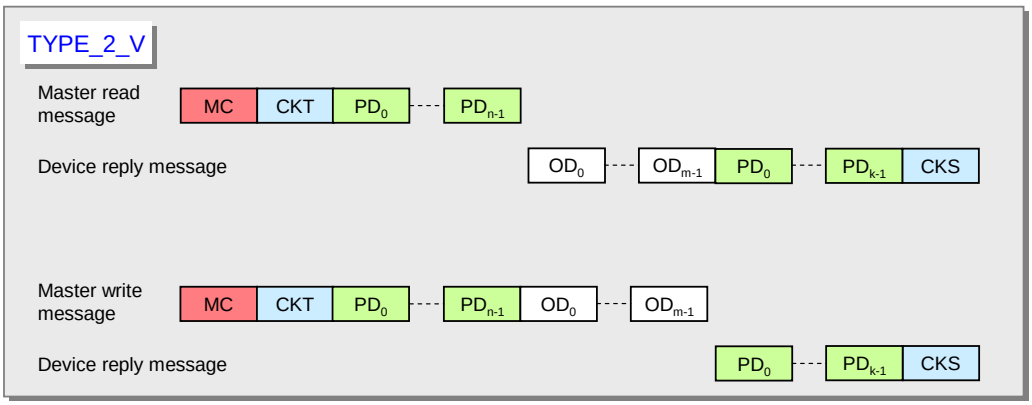


Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.14 – Séquence M de TYPE_2_6

La séquence M de TYPE_2_V transmet, par cycle, l'ensemble n (k) octets de ProcessDataIn en écriture (lecture). La plage de valeurs de n (k) est de 0 à 32. Il n'y a pas de PDin (PD d'entrée) ou de PDout (PD de sortie) lorsque n = 0 ou k = 0. Le TYPE_2_V transmet également, par cycle, m octets de Données à la Demande de lecture ou d'écriture (segmentées) en utilisant l'adresse donnée en Figure A.1. Les Valeurs autorisées pour m sont 1, 2, 8 et 32. Ce type de séquence M variable est illustré en Figure A.15.



Légende

Anglais	Français
Master read message	Message de lecture du Maître
Device reply message	Message de réponse du Dispositif
Master write message	Message d'écriture du Maître

Figure A.15 – Séquence M de TYPE_2_V

Pour l'accès en écriture des Données à la Demande via les canaux de communication de page et de diagnostic, seul est évalué le premier octet (OD_0) des Données à la Demande. Le Dispositif doit ignorer les octets restants. Le Maître doit envoyer tous les autres OD à "0".

A.2.5 Séquence M de TYPE _3

La séquence M de TYPE _3 est réservée et ne doit pas être utilisée.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.2.6 Utilisation des types de séquence M pour les modes STARTUP, PREOPERATE et OPERATE

Le Tableau A.7 énumère les types de séquence M pour le mode STARTUP ainsi que le temps de récupération minimal (T_{initcyc}) qui doit être observé pour les mises en œuvre du Maître (voir A.3.9). Le code de séquence M fait référence au codage défini en B.1.4.

Tableau A.7 – Types de séquence M pour le mode STARTUP

STARTUP code de séquence M	Données à la demande	Type de séquence M	Temps de récupération minimal
	Octets		T_{BIT}
n/a	1	TYPE_0	100

Le Tableau A.8 énumère les types de séquence M pour le mode PREOPERATE ainsi que le temps de récupération minimal (T_{initcyc}) qui doit être observé pour les mises en œuvre du Maître.

Tableau A.8 – Types de séquence M pour le mode PREOPERATE

PREOPERATE code de séquence M	Données à la demande	Type de séquence M	Temps de récupération minimal
	Octets		T_{BIT}
0	1	TYPE_0	100
1	2	TYPE_1_2	100
2	8	TYPE_1_V	210
3	32	TYPE_1_V	550
NOTE Le temps de récupération minimal en mode PREOPERATE est une exigence pour le Maître.			

Le Tableau A.9 énumère les types de séquence M du mode OPERATE pour des Dispositifs patrimoniaux. La durée de cycle minimale pour le Maître en mode OPERATE est spécifiée par le paramètre "MinCycleTime" du Dispositif (voir B.1.3).

Tableau A.9 – Types de séquence M pour le mode OPERATE (protocole héritage)

OPERATE code de séquence M	Données à la demande	Données de processus (PD)		Type de séquence M
	Octets	PDin	PDout	Protocole patrimonial (voir [8])
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
sans effet	2	3…32 octets	0…32 octets	TYPE_1_1/1_2 (entrelacés)
sans effet	2	0…32 octets	3…32 octets	TYPE_1_1/1_2 (entrelacés)
sans effet	1	1…8 bits	0	TYPE_2_1
sans effet	1	9…16 bits	0	TYPE_2_2
sans effet	1	0	1…8 bits	TYPE_2_3
sans effet	1	0	9…16 bits	TYPE_2_4
sans effet	1	1…8 bits	1…8 bits	TYPE_2_5

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau A.10 énumère les types de séquence M du mode OPERATE pour des Dispositifs conformes à la présente norme. La durée de cycle minimale pour le Maître en mode OPERATE est spécifiée par le paramètre "MinCycleTime" du Dispositif (voir B.1.3).

Tableau A.10 – Types de séquence M pour le mode OPERATE

OPERATE code de séquence M	Données à la demande	Données de processus (PD)		Type de séquence M
	Octets	PDin	PDout	
0	1	0	0	TYPE_0
1	2	0	0	TYPE_1_2
6	8	0	0	TYPE_1_V
7	32	0	0	TYPE_1_V
0	2	3…32 octets	0…32 octets	TYPE_1_1/1_2 entrelacés
0	2	0…32 octets	3…32 octets	TYPE_1_1/1_2 entrelacés
0	1	1…8 bits	0	TYPE_2_1
0	1	9…16 bits	0	TYPE_2_2
0	1	0	1…8 bits	TYPE_2_3
0	1	0	9…16 bits	TYPE_2_4
0	1	1…8 bits	1…8 bits	TYPE_2_5
0	1	9…16 bits	1…16 bits	TYPE_2_6
0	1	1…16 bits	9…16 bits	TYPE_2_6
4	1	0…32 octets	3…32 octets	TYPE_2_V
4	1	3…32 octets	0…32 octets	TYPE_2_V
5	2	>0 bit, octets	≥0 bit, octets	TYPE_2_V
5	2	≥0 bit, octets	>0 bit, octets	TYPE_2_V
6	8	>0 bit, octets	≥0 bit, octets	TYPE_2_V
6	8	≥0 bit, octets	>0 bit, octets	TYPE_2_V
7	32	>0 bit, octets	≥0 bit, octets	TYPE_2_V
7	32	≥0 bit, octets	>0 bit, octets	TYPE_2_V

A.3 Contraintes temporelles

A.3.1 Généralités

Les interactions d'un Maître et de son Dispositif sont caractérisées par plusieurs contraintes temporelles qui s'appliquent aux temps de transmission de la trame UART et de messages du Maître et du Dispositif, complétés par des temps de réponse, de cycle, de retard et de récupération.

A.3.2 Temps binaire

Le temps binaire T_{BIT} est la durée nécessaire à la transmission d'un bit unique. Sa valeur est l'inverse de la vitesse de transmission (voir l'Equation (A.2)).

$$T_{BIT} = 1/(\text{vitesse de transmission}) \quad \text{(A.2)}$$

Les valeurs de T_{BIT} sont spécifiées dans le Tableau 8.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.3.3 Retard de transmission de trame UART de Maître (ports)

Le retard de transmission de la trame UART t_1 d'un port est la durée de la pause entre la fin du bit d'arrêt d'une trame UART et le début du bit de départ de la trame UART suivante. Le port doit transmettre les trames UART dans un délai maximal de pause de 1 temps binaire (voir l'Equation (A.3)).

$$0 \leq t_1 \leq 1\ T_{BIT} \quad \text{(A.3)}$$

A.3.4 Retard de transmission de trame UART de Dispositifs

Le retard de transmission de la trame UART t_2 d'un Dispositif est la durée de la pause entre la fin du bit d'arrêt d'une trame UART et le début du bit de départ de la trame UART suivante. Le Dispositif doit transmettre les trames UART dans un délai maximal de pause de 3 temps binaires (voir l'Equation (A.4)).

$$0 \leq t_2 \leq 3\ T_{BIT} \quad \text{(A.4)}$$

A.3.5 Temps de réponse de Dispositifs

Le temps de réponse de Dispositif t_A est la durée de la pause entre la fin du bit d'arrêt de la dernière trame UART à recevoir d'un Port et le début du bit de départ de la première trame UART envoyée. Le Dispositif doit observer une pause d'au moins un temps binaire sans dépasser 10 temps binaires (voir l'Equation (A.5)).

$$1\ T_{BIT} \leq t_A \leq 10\ T_{BIT} \quad \text{(A.5)}$$

A.3.6 Temps de séquence M

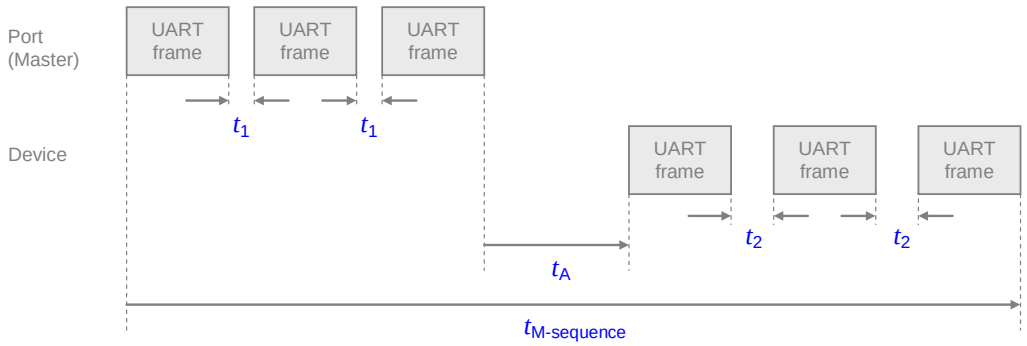
La communication entre un port et le Dispositif qui lui est associé a lieu selon une planification fixe appelée temps de séquence M (voir l'Equation (A.6)).

$$t_{\text{séquence M}} = (m+n) \times 11 \times T_{BIT} + t_A + (m\text{-}1) \times t_1 + (n\text{-}1) \times t_2 \quad \text{(A.6)}$$

Dans cette formule, m est le nombre de trames UART envoyées par le port au Dispositif et n est le nombre de trames UART envoyées par le Dispositif au port. La formule peut uniquement être utilisée pour des estimations, car les temps t_1 et t_2 peuvent ne pas être constants.

La Figure A.16 représente les temporisations d'une séquence M constituée d'un message (de port) de Maître et d'un message de Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Port (Master)	Port (Maître)
Device	Dispositif
UART frame	Trame UART
$t_{M\text{-sequence}}$	$t_{séquence\ M}$

Figure A.16 – Temporisation de séquence M

A.3.7 Durée de cycle

La durée d'un cycle t_{CYC} (voir l'Equation (A.7)) dépend du paramètre de Dispositif "MinCycleTime" ainsi que de la conception et de la mise en œuvre d'un Maître et du nombre de ports.

$$t_{CYC} = t_{séquence\ M} + t_{idle} \quad (A.7)$$

Le paramètre réglable du Dispositif "MasterCycleTime" peut être utilisé pour la conception d'une technologie spécifique au Dispositif, un actionneur par exemple, pour en déduire les conditions de temporisation d'une action par défaut appropriée telle que la désactivation ou la mise hors tension de l'actionneur (voir 7.3.3.5 "MaxCycleTime", 10.2, et 10.7.3).

Le Tableau A.11 énumère les valeurs minimales recommandées de durée de cycle à utiliser pour le mode de transmission spécifié d'un port. Les valeurs sont calculées sur la base de la séquence M de Type_2_1.

Tableau A.11 – Valeurs recommandées de MinCycleTimes

Mode d'émission	t_{CYC}
COM1	18,0 ms
COM2	2,3 ms
COM3	0,4 ms

A.3.8 Temps au repos

Le temps au repos t_{idle} résulte de la durée de cycle configurée t_{CYC} et du temps de séquence M $t_{séquence\ M}$. En référence à un port donné, il comprend la durée entre la fin du message d'un Dispositif et le début du message suivant provenant du Maître (port).

Le temps au repos doit être suffisamment long pour que le Dispositif soit prêt à recevoir le message suivant.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.3.9 Temps de récupération

Le Maître doit attendre pendant un temps de récupération $t_{initcyc}$ entre n'importe lequel des deux accès acycliques successifs aux Dispositifs en phase STARTUP ou PREOPERATE (voir A.2.6).

A.4 Erreurs et mesures correctives

A.4.1 Erreurs d'UART

A.4.1.1 Erreurs de parité

Le bit de parité UART (voir la Figure 18) et la somme de contrôle (voir A.1.6) sont deux mécanismes indépendants permettant de garantir le transfert de données. Cela signifie, par exemple, que deux erreurs binaires dans différents octets d'un message, donnant lieu à une somme de contrôle correcte, peuvent également être détectées. Les deux mécanismes donnent lieu au même traitement d'erreur.

Mesure corrective: Le Maître doit répéter le message de Maître 2 fois (voir 7.2.2.1). Les Dispositifs doivent rejeter toutes les données pour lesquelles des erreurs ont été détectées et ne pas générer de réaction.

A.4.1.2 Erreurs de verrouillage de trame UART

Les conditions de détection correcte d'une trame UART sont spécifiées en 5.3.3.2. Le traitement des erreurs doit avoir lieu chaque fois que des formes de signal perturbées ou des temporisations incorrectes entraînent un bit d'arrêt de trames UART non valide.

Mesure corrective: Voir A.4.1.1.

A.4.2 Erreurs de réactivation

L'impulsion de courant de réactivation est spécifiée en 5.3.3.3 et les procédures de réactivation en 7.3.2.1. Plusieurs défauts peuvent apparaître pendant des tentatives d'établissement de la communication.

Mesure corrective: de nouvelles tentatives sont possibles. Une description plus détaillée est fournie en 7.3.2.1.

A.4.3 Erreurs de transmission

A.4.3.1 Erreurs de somme de contrôle

Le mécanisme de calcul de la somme de contrôle est spécifié en A.1.6. Toute erreur de somme de contrôle donne lieu à un traitement d'erreur.

Mesure corrective: Voir A.4.1.1.

A.4.3.2 Erreurs d'expiration du délai

Les diverses contraintes temporelles liées aux séquences M sont spécifiées en A.3. Le Maître (ports) et les Dispositifs vérifient plusieurs temporisations critiques, telles que l'absence de synchronisme dans les messages.

Mesure corrective: Voir A.4.1.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.4.3.3 Collisions

Il y a collision lorsque, suite à une erreur, le Maître et le Dispositif émettent simultanément. Cette erreur est interprétée comme une séquence M défectueuse.

Mesure corrective: Voir A.4.1.1.

A.4.4 Erreurs de protocole

Une erreur de protocole a lieu lorsque, par exemple, la séquence de transmission segmentée d'une ISDU est erronée (voir le cas de contrôle de flux en A.1.2).

Mesure corrective: abandon du service avec une information ErrorType (voir l'Annexe C).

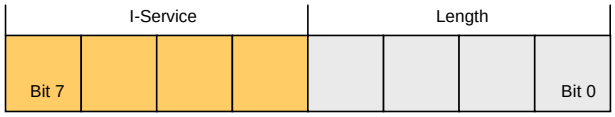
A.5 Structure générale et codage de ISDU

A.5.1 Aperçu

L'objectif et la structure générale d'une ISDU sont spécifiés en 7.3.6.1. Les paragraphes A.5.2 à A.5.7 fournissent une description détaillée des différents éléments d'une ISDU, ainsi que certains exemples.

A.5.2 "Service I"

La Figure A.17 présente la structure de l'octet de "Service I".



Légende

Anglais	Français
I-Service	"Service I"
Length	Longueur

Figure A.17 – Octet de "Service I"

Bits 0 à 3: Longueur
Le codage de la longueur réduite (quartet) de l'ISDU est spécifié dans le Tableau A.14 .

Bits 4 à 7: "Service I"
Le codage du "Service I" réduit de l'ISDU est spécifié dans le Tableau A.12.

Tous les autres éléments de la structure spécifiée en 7.3.6.1 sont transmis comme des octets indépendants.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau A.12 – Définition du "Service I" réduit (quartet)

"Service I" (binaire)	Définition		Format des index
	Maître	Dispositif	
0000	Pas de Service	Pas de Service	n/a
0001	Demande d'écriture	Réservé	Index sur 8 bits
0010	Demande d'écriture	Réservé	Index et Sous-index sur 8 bits
0011	Demande d'écriture	Réservé	Index et Sous-index sur 16 bits
0100	Réservé	Réponse d'écriture (-)	aucun
0101	Réservé	Réponse d'écriture (+)	aucun
0110	Réservé	Réservé	
0111	Réservé	Réservé	
1000	Réservé	Réservé	
1001	Demande de lecture	Réservé	Index sur 8 bits
1010	Demande de lecture	Réservé	Index et Sous-index sur 8 bits
1011	Demande de lecture	Réservé	Index et Sous-index sur 16 bits
1100	Réservé	Réponse de lecture (-)	aucun
1101	Réservé	Réponse de lecture (+)	aucun
1110	Réservé	Réservé	
1111	Réservé	Réservé	

Le Tableau A.13 définit la syntaxe des ISDU. Le ErrorType (type d'erreur) est donné dans l'Annexe C.

Tableau A.13 – Syntaxe des ISDU

Nom de l'ISDU	Structure de l'ISDU
Demande d'écriture	{I-Service(0x1), LEN, Index, [Data*], CHKPDU} ^ {I-Service(0x2), LEN, Index, Subindex, [Data*], CHKPDU} ^ {I-Service(0x3), LEN, Index, Index, Subindex, [Data*], CHKPDU}
Réponse d'écriture (+)	I-Service(0x5), Length(0x2), CHKPDU
Réponse d'écriture (-)	I-Service(0x4), Length(0x4), ErrorType, CHKPDU
Demande de lecture	{I-Service(0x9), Length(0x3), Index, CHKPDU} ^ {I-Service(0xA), Length(0x4), Index, Subindex, CHKPDU} ^ {I-Service(0xB), Length(0x5), Index, Index, Subindex, CHKPDU}
Réponse de lecture (+)	I-Service(0xD), LEN, [Data*], CHKPDU
Réponse de lecture (-)	I-Service(0xC), Length(0x4), ErrorType, CHKPDU
Légende LEN = {Length(0x1), ExtLength} ^ {Length}	

A.5.3 Longueur étendue (ExtLength)

Le nombre d'octets transmis dans ce "Service I", y compris l'ensemble des informations de protocole (6 octets), est spécifié dans l'élément “Longueur” d'une ISDU. Si la longueur totale est supérieure à 15 octets, la longueur est spécifiée en utilisant les informations de longueur

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

étendue ("ExtLength"). Les valeurs admissibles pour “Length” et "ExtLength" sont données dans le Tableau A.14.

Tableau A.14 – Définition de la longueur réduite (quartet) et de la longueur étendue d'octet ExtLength

"Service I"	Longueur	ExtLength	Définition
0	0	n/a	Pas de service, la longueur de l'ISDU est 1. Utilisation du protocole.
0	1	n/a	Dispositif occupé, la longueur de l'ISDU est 1. Utilisation du protocole.
0	2 à 15	n/a	Réservé et ne doit pas être utilisé
1 à 15	0	n/a	Réservé et ne doit pas être utilisé
1 à 15	1	0 à 16	Réservé et ne doit pas être utilisé
1 à 15	1	17 à 238	Longueur de l'ISDU en "ExtLength"
1 à 15	1	239 à 255	Réservé et ne doit pas être utilisé
1 à 15	2 à 15	n/a	Longueur de l'ISDU

A.5.4 Index et Sous-index

L'adresse de paramètre de l'objet de données à transmettre au moyen de l'ISDU est spécifiée dans l'élément “Index”. La plage de valeurs d'“Index” va de 0 à 65 535 (voir B.2.1 pour les contraintes). Les valeurs d'Index 0 et 1 doivent être rejetées par le Dispositif.

Il est inutile que le Dispositif prenne en charge toutes les valeurs d'Index et de Sous-index. Le Dispositif doit envoyer une réponse négative aux valeurs d'Index ou de Sous-index non prises en charge.

L'adresse des éléments de données d'un paramètre structuré de l'objet de données à transmettre au moyen de l'ISDU est spécifiée dans l'élément “Sous-index”. La plage de valeurs de “Sous-index” va de 0 à 255 et une valeur "0" est utilisée pour référencer l'ensemble de l'objet de données (voir la Figure 5).

Le Tableau A.15 donne les formats d'Index utilisés dans l'ISDU en fonction des paramètres transmis.

Tableau A.15 – Utilisation des formats d'Index

Index	Sous-index	Format d'Index d'ISDU
0 à 255	0	Index sur 8 bits
0 à 255	1 a 255	Index sur 8 bits et Sous-index sur 8 bits
256 à 65 535	0 à 255	Index sur 16 bits et Sous-index sur 8 bits (voir NOTE)
NOTE Voir B.2.1 pour les contraintes concernant la plage de valeur des Index		

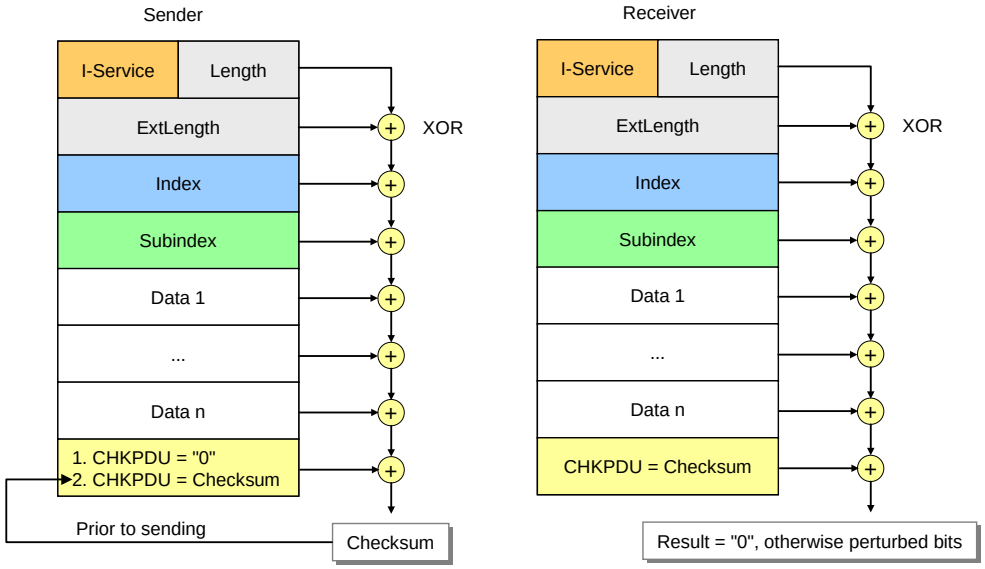
A.5.5 Données

L'élément “Données” peut contenir les objets de données spécifiés en Annexe B ou des objets de données spécifiques au Dispositif. La longueur des données correspond aux entrées dans l'élément “Longueur” moins les éléments de protocole ISDU.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.5.6 Vérification des ISDU (CHKPDU)

L'élément “CHKPDU” assure l'intégrité des données. L'émetteur calcule la valeur de "CHKPDU" par traitement XOR de tous les octets d'une ISDU, y compris "CHKPDU" avec une valeur provisoire "0", qui est ensuite remplacée par le résultat du calcul (voir Figure A.18).



Légende

Anglais	Français
Sender	Emetteur
Receiver	Destinataire
I-Service	"Service I"
Length	Longueur
ExtLength	ExtLength (longueur étendue)
Subindex	Sous-index
Data	Donnée
CHKPDU = Checksum	CHKPDU = Somme de contrôle
Prior to sending	Avant envoi
Checksum	Somme de contrôle
Result = "0", otherwise perturbed bits	Résultat = "0", sinon bits perturbés

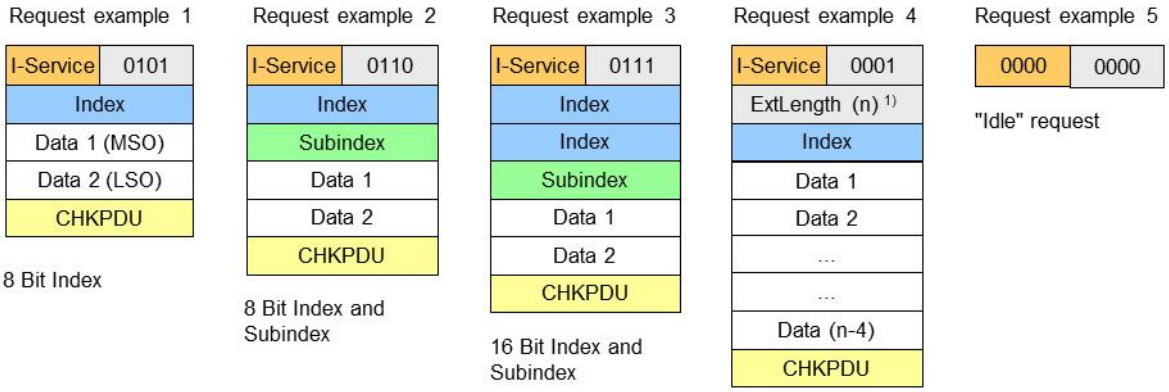
Figure A.18 – Vérification de l'intégrité des ISDU via CHKPDU

Le destinataire vérifie que le traitement XOR de tous les octets de l'ISDU donne bien le résultat "0" (voir la Figure A.18). Si le résultat est différent de "0", un traitement d'erreur doit avoir lieu. Voir également A.1.6.

A.5.7 Exemples d'ISDU

La Figure A.19 représente des exemples types de formats de demande d'ISDU; ceux-ci sont expliqués dans les alinéas suivants.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



1) ExtLength globale d'ISDU = *n* (1 à 238); Longueur = 1 ("0001")

Légende

Anglais	Français
Request example 1	Exemple de demande 1
Request example 2	Exemple de demande 2
Request example 3	Exemple de demande 3
Request example 4	Exemple de demande 4
Request example 5	Exemple de demande 5
I-Service	"Service I"
Data 1 (MSO)	Données 1 (MSO)
Data 2 (LSO)	Données 2 (LSO)
8 Bit Index	Index sur 8 bits
Subindex	Sous-index
8 Bit Index and Subindex	Index et Sous-index sur 8 bits
16 Bit Index and Subindex	Index et Sous-index sur 16 bits
"Idle" request	Demande "Repos"

Figure A.19 – Exemples de formats de demande d'ISDU

La demande d'ISDU de l'exemple 1 comprend un élément "Index" permettant un adressage de 0 à 254 (voir le Tableau A.15). Dans cet exemple, le sous-index est égal à "0" et le contenu total de "Index" est Données 1 avec l'octet de poids fort (MSO) et Données 2 avec l'octet de poids faible (LSO). Sa longueur totale est 5 ("0101").

La demande d'ISDU de l'exemple 2 comprend un élément "Index" permettant un adressage de 0 à 254 et l'élément "Sous-index" permettant l'adressage d'un élément de structure de données. Sa longueur totale est 6 ("0110").

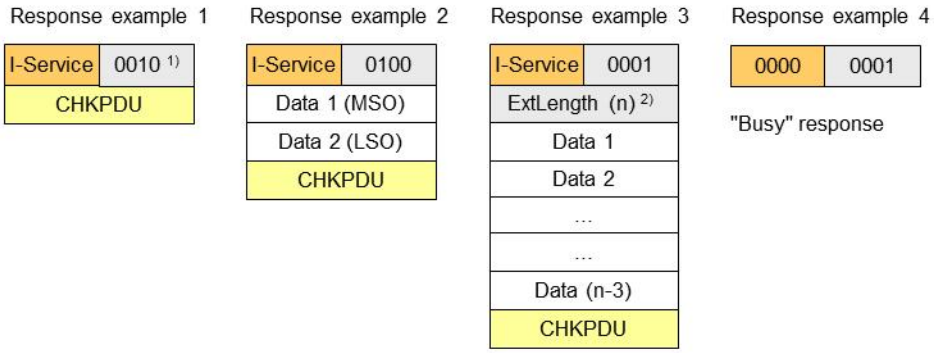
La demande d'ISDU de l'exemple 3 comprend deux éléments "Index" permettant un adressage de 256 à 65 535 (voir le Tableau A.15) et l'élément "Sous-index" permettant l'adressage d'un élément de structure de données. Sa longueur totale est 7 ("0111").

La demande d'ISDU de l'exemple 4 comprend un élément "Index" et l'élément "ExtLength" indiquant le nombre d'éléments d'ISDU (*n*, de 17 à 238). Dans ce cas l'élément, "longueur" prend la valeur "1".

La demande d'ISDU "Repos" dans l'exemple 5 est utilisée pour indiquer qu'aucun service n'est en attente.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Figure A.20 représente des exemples types de ISDU de réponse, expliqués dans les alinéas suivants.



1) Longueur minimale = 2 ("0010")
2) ExtLength globale d'ISDU = *n* (17 à 238);
Longueur = 1 ("0001")

Légende

Anglais	Français
Response example 1	Exemple de réponse 1
Response example 2	Exemple de réponse 2
Response example 3	Exemple de réponse 3
Response example 4	Exemple de réponse 4
I-Service	"Service I"
Data 1 (MSO)	Données 1 (MSO)
Data 2 (LSO)	Données 2 (LSO)
ExtLength (n)[2)]	ExtLength (n)[2)]
Data	Donnée
"Busy" response	Réponse "Occupé"

Figure A.20 – Exemples de ISDU de réponse

La ISDU de réponse de l'exemple 1 présente une valeur minimale de 2 pour l'élément "Longueur" ("0010").

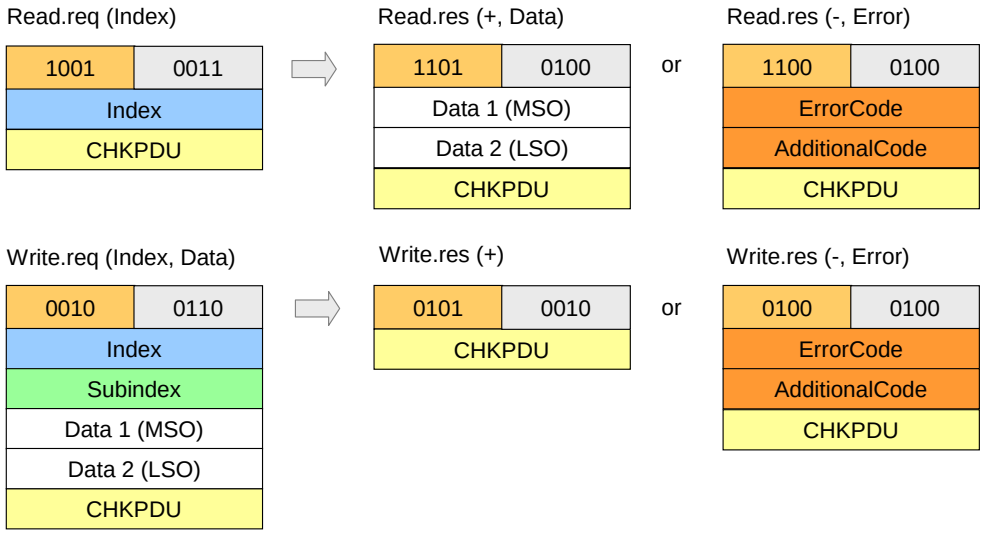
La ISDU de réponse de l'exemple 2 présente deux éléments de données et un nombre total de 4 éléments dans l'élément "Longueur" ("0100"). L'élément "Données 1" transporte l'octet de poids fort (MSO) et l'élément "Données 2" l'octet de poids faible (LSO).

La ISDU de réponse de l'exemple 3 présente l'élément "ExtLength" indiquant le nombre d'éléments d'ISDU (*n*, de 17 à 238). Dans ce cas l'élément "Longueur" prend la valeur "1".

La ISDU de réponse "Occupé" de l'exemple 4 est utilisée lorsqu'un Dispositif est actuellement incapable de répondre à la demande de lecture du Maître du fait du délai nécessaire de préparation pour la réponse.

La Figure A.21 illustre des exemples types de ISDU de demande de lecture et d'écriture, expliqués dans les alinéas suivants.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
(+, Data)	(+, Données)
(-, Error)	(-, Erreur)
Data 1 (MSO)	Données 1 (MSO)
Data 2 (LSO)	Données 2 (LSO)
(Index, Data)	(Index, Données)
Subindex	Sous-index

Figure A.21 – Exemples de ISDU de demande de lecture et d'écriture

Le code de la demande de lecture "Service I" est "1001". Comme décrit dans le Tableau A.13, il comprend un élément "Index". Une réponse de lecture réussie (+) du Dispositif avec le code "1101" est présentée à côté de la demande avec deux éléments de données. Sa longueur totale est 4 ("0100"). Un échec de la réponse de lecture (-) du Dispositif avec le code "1100" est présenté sur la ligne suivante. Il contient le ErrorType avec les deux éléments de données ErrorCode et AdditionalCode (voir l'Annexe C).

Le code de la demande d'écriture "Service I" est "0010". Comme décrit dans le Tableau A.13, il comprend un élément "Index" et un élément "Sous-index". Une réponse d'écriture réussie (+) du Dispositif avec le code "0101" est présentée à côté de la demande sans éléments de données. Sa longueur totale est 2 ("0010"). Un échec de la réponse de lecture (-) du Dispositif avec le code "0100" est présenté sur la ligne suivante. Il contient le ErrorType avec les deux éléments de données ErrorCode et AdditionalCode (voir l'Annexe C).

A.6 Structure générale et codage d'Événements

A.6.1 Généralités

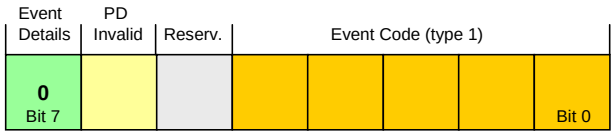
L'objectif et la structure générale de la mémoire d'Événements sont spécifiés au 7.3.8.1 et dans le Tableau 56. Cette mémoire inclut un Code d'État (StatusCode), plusieurs Qualificatifs d'Événement (EventQualifiers) et les Codes d'Événements (EventCodes) correspondants. Le codage de ces éléments de mémoire est spécifié de A.6.2 à A.6.5.

A.6.2 StatusCode de type 1 (sans détails)

La Figure A.22 illustre la structure de ce StatusCode.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

NOTE 1 Le StatusCode de type 1 est uniquement utilisé dans des Événements générés par des dispositifs patrimoniaux (voir 7.3.8.1).



Légende

Anglais	Français
Event Details	Détails d'événements
PD Invalid	PD non valides
Reserv.	Réservé.
Event Code (type 1)	Code événement (type 1)

Figure A.22 – Structure du StatusCode de type 1

Bits 0 à 4: EventCode (type 1)
Le codage de cette structure de données est présenté dans le Tableau A.16. Les EventCodes sont mis en correspondance dans des EventCodes (de type 2) comme décrit en Annexe D. Voir 7.3.8.2 pour plus d'informations.

Tableau A.16 – Mise en correspondance des EventCodes (type 1)

EventCode (type 1)	EventCode (type2)	Instance	Type	Mode
****1	0xFF80	Application	Notification	Occurrence unique d'événement
***1*	0xFF80	Application	Avertissement	Occurrence unique d'événement
1	0x6320	Application	Erreur	Occurrence unique d'événement
*1***	0xFF80	Application	Erreur	Occurrence unique d'événement
1****	0xFF10	Application	Erreur	Occurrence unique d'événement
Légende * Sans effet type 2 Voir le Tableau D.1 et le Tableau D.2				

Bit 5: Réservé
Ce bit est réservé et doit être mis à zéro dans le StatusCode de type 1.

Bit 6: Réservé

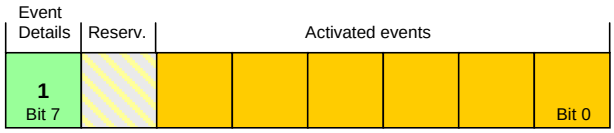
NOTE 2 Ce bit est utilisé dans un protocole héritage (voir [8]) pour l'indication PDinvalid.

Bit 7: Détails d'événements
Ce bit indique qu'aucune information détaillée d'Événement n'est disponible. Il doit toujours être mis à zéro dans le StatusCode de type 1.

A.6.3 StatusCode de type 2 (avec détails)

La Figure A.23 illustre la structure du StatusCode de type 2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



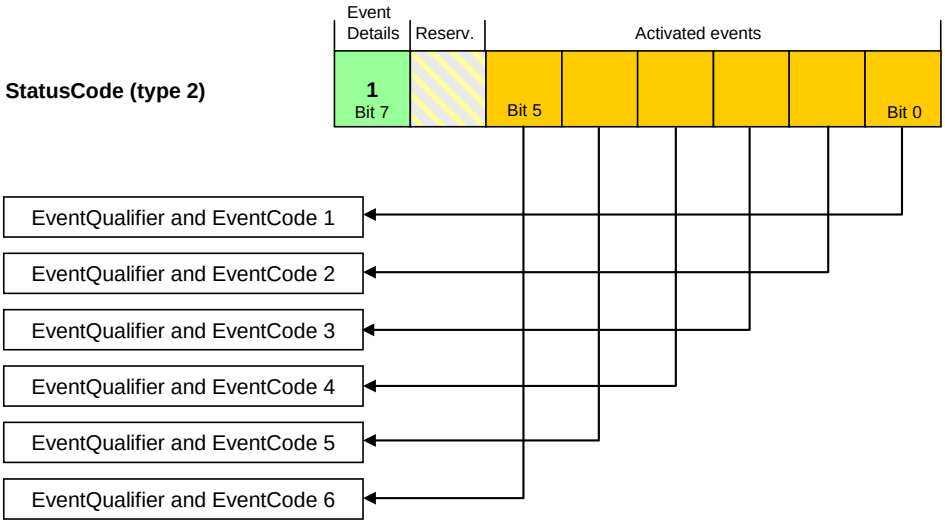
Légende

Anglais	Français
Event Details	Détails d'événements
Reserv.	Réservé.
Activated events	Événements activés

Figure A.23 – Structure du StatusCode de type 2

Bits 0 à 5: Événements activés

Chaque bit est associé à un Événement dans la mémoire (voir 7.3.8.1), comme le montre la Figure A.24. Le bit 0 est associé à l'Événement 1, le bit 1 à l'Événement 2, etc. Un bit ayant la valeur "1" indique que les formats du EventQualifier correspondant et du EventCode saisis dans la mémoire sont valides. Un bit ayant la valeur "0" indique une entrée non valide.



Légende

Anglais	Français
Event.Details	Détails d'événements
Reserv.	Réservé.
Activated events	Événements activés
StatusCode (type 2)	Code d'État (type 2)
EventQualifier and EventCode	Qualificatif d'Événement et Code d'Événement

Figure A.24 – Indication d'Événements activés

Bit 6: Réservé

Ce bit est réservé et doit être mis à zéro.

NOTE Ce bit est utilisé dans la version de protocole héritage (voir [8]) pour l'indication PDinvalid.

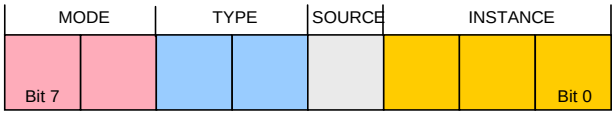
Bit 7: Détails d'événements

Ce bit indique que des informations détaillées d'Événement sont disponibles. Il doit toujours être établi dans le StatusCode de type 2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

A.6.4 EventQualifier

La structure du EventQualifier est illustrée en Figure A.25.



Légende

Anglais	Français
SOURCE	ORIGINE

Figure A.25 – Structure du EventQualifier

Bits 0 à 2: INSTANCE
Ces bits indiquent l'origine particulière (instance) d'un Événement permettant ainsi une évaluation plus fine du côté du destinataire. Les valeurs admissibles de "INSTANCE" sont énumérées dans le Tableau A.17.

Tableau A.17 – Valeurs de "INSTANCE"

Valeur	Définition
0	Inconnu
1 à 3	Réservé
4	Application
5 à 7	Réservé

Bit 3: SOURCE (origine)
Ce bit indique l'origine de l'Événement. Les valeurs admissibles de "SOURCE" sont énumérées dans le Tableau A.18.

Tableau A.18 – Valeurs de "SOURCE"

Value	Définition
0	Dispositif (distant)
1	Maître (local)

Bits 4 à 5: TYPE
Ces bits indiquent la catégorie de l'Événement. Les valeurs admissibles de "TYPE" sont énumérées dans le Tableau A.19.

Tableau A.19 – Valeurs de "TYPE"

Valeur	Définition
0	Réservé
1	Notification
2	Avertissement
3	Erreur

Bits 6 à 7: MODE
Ces bits indiquent le mode de l'Événement. Les valeurs admissibles de "MODE" sont énumérées dans le Tableau A.20.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau A.20 – Valeurs de "MODE"

Valeur	Définition
0	Réservé
1	Occurrence unique d'événement
2	Disparition d'Événement
3	Apparition d'Événement

A.6.5 EventCode

L'entrée EventCode contient l'identifiant d'un Événement réel. Les valeurs admissibles de "EventCode" sont énumérées en Annexe D.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe B
(normative)

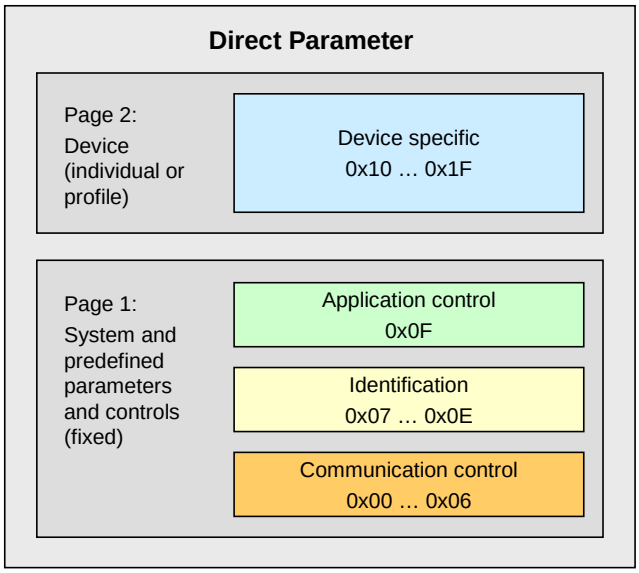
Paramètres et commandes

B.1 Pages 1 et 2 de Paramètres Directs

B.1.1 Aperçu

Comme illustré en Figure 5, le concepteur d'un Dispositif peut en général utiliser beaucoup d'espace pour les paramètres et les commandes. Cependant, les petits capteurs ayant un nombre limité de paramètres et de ressources recherchent un sous-ensemble simple. L'interface SDCI met à disposition ce qu'on appelle les pages 1 et 2 de Paramètres Directs avec une méthode d'accès simplifiée (canal de communication de page conformément au Tableau A.1) pour répondre à cette exigence.

La gamme de Paramètres Directs est structurée comme illustré en Figure B.1. Les paramètres directs sont divisés en page 1 et page 2.



Légende

Anglais	Français
Direct Parameter	Paramètres Directs
Device (individual or profile)	Dispositif (individuel ou profil)
Device specific	Spécifique au Dispositif
0x10 … 0x1F	0x10 à 0x1F
System and predefined parameters and controls (fixed)	Système ainsi que paramètres et contrôles prédéfinis (fixes)
Application control	Contrôle des applications
Identification	Identification
0x07 … 0x0E	0x07 à 0x0E
Communication control	Contrôle de la communication
0x00 … 0x06	0x00 à 0x06

Figure B.1 – Classification et mise en correspondance des Paramètres Directs

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La Page 1 va de 0x00 à 0x0F. Elle comprend les catégories de paramètres suivantes:

- contrôle de la communication;
- paramètre d'identification;
- contrôle des applications.

La couche d'application (AL) du Maître permet un accès en lecture seule à la page 1 de Paramètres Directs sous la forme d'objets de données (voir 8.2.1) via l'Index 0. Chaque octet individuel peut être lu via l'Index 0 et le Sous-index correspondant. Le Sous-index 1 indique l'adresse 0x00 et le Sous-index 16 indique l'adresse 0x0F.

La Page 2 va de 0x10 à 0x1F. Cette page comprend les paramètres qui sont utilisés de manière facultative par la technologie du Dispositif particulier. La couche d'application (AL) du Maître permet un accès en lecture/écriture à la page 2 de Paramètres Directs sous la forme d'objets de données (voir 8.2.1) via l'Index 1. Chaque octet individuel peut être écrit ou lu via l'Index 1 et le Sous-index correspondant. Le Sous-index 1 indique l'adresse 0x10 et le Sous-index 16 indique l'adresse 0x1F.

Un Dispositif doit toujours renvoyer la valeur "0" suite à un accès en lecture aux adresses de paramètres directs, qui ne sont pas mises en œuvre (par exemple en cas d'adresses de paramètres réservés ou de paramètres facultatifs non pris en charge). Le Dispositif doit ignorer un accès en écriture à des paramètres non utilisés.

La structure des pages 1 et 2 de Paramètres Directs est spécifiée dans le Tableau B.1.

Tableau B.1 – Pages 1 et 2 de Paramètres Directs

Adresse	Nom du paramètre	Accès	Mise en œuvre/référence	Description
Page 1 de Paramètres Directs				
0x00	Master-Command	W (écriture)	Obligatoire/ voir B.1.2	Commande du Maître pour basculer entre états opérationnels (voir la NOTE 1)
0x01	MasterCycle-Time	R/W (Lecture/ Écriture)	Obligatoire/ voir B.1.3	Durée de cycle réelle utilisée par le Maître pour solliciter le Dispositif. Peut être utilisée comme un paramètre de surveillance du transfert des Données de Processus.
0x02	MinCycleTime	R (lecture)	Obligatoire/ voir B.1.3	Durée de cycle minimale prise en charge par un Dispositif. Il s'agit d'une caractéristique de performance du Dispositif qui dépend de sa technologie et de sa mise en œuvre.
0x03	M-sequence Capability (Capacité de séquence M)	R (lecture)	Obligatoire/ voir B.1.4	Informations concernant la mise en œuvre d'options relative aux séquences M et à la configuration physique
0x04	RevisionID	R/W (Lecture/ Écriture)	Obligatoire/ voir B.1.5	ID de la version de protocole utilisée pour la mise en œuvre (doit être mis à 0x11)
0x05	ProcessDataIn	R (lecture)	Obligatoire/ voir B.1.6	Nombre et structure des données d'entrée (Données de Processus du Dispositif vers le Maître)
0x06	ProcessData-Out	R (lecture)	Obligatoire/ voir B.1.7	Nombre et structure des données de sortie (Données de Processus du Maître vers le Dispositif)
0x07	VendorID 1 (MSB)	R (lecture)	Obligatoire/ voir B.1.8	Identification unique du fournisseur (voir la NOTE 2)
0x08	VendorID 2 (LSB)			
0x09	DeviceID 1 (Octet 2, MSB)	R/W (Lecture/	Obligatoire/ voir B.1.9	Identification unique du Dispositif allouée par un fournisseur donné

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Adresse	Nom du paramètre	Accès	Mise en œuvre/référence	Description
0x0A	DeviceID 2 (Octet 1)	Écriture)		
0x0B	DeviceID 3 (Octet 0, LSB)			
0x0C	FunctionID 1 (MSB)	R (lecture)	voir B.1.10	Réservé (le personnel technique doit mettre les deux octets à "0x00")
0x0D	FunctionID 2 (LSB)			
0x0E		R	Réservé	
0x0F	Commande système	W (écriture)	Facultatif/ voir B.1.11	Interface de commande pour les applications de l'utilisateur uniquement et les Dispositifs sans prise en charge des ISDU (voir NOTE 1)
Page 2 de Paramètres Directs				
0x10 à 0x1F	Spécifique au fournisseur	Faculta-tif	Facultatif/ voir B.1.12	Paramètres spécifiques au Dispositif
NOTE 1 Une opération de lecture renvoie des valeurs non spécifiées. NOTE 2 Les VendorID (identifiants de fournisseurs) sont attribués par le "IO-Link consortium".				

B.1.2 MasterCommand

L'application du Maître est capable de vérifier l'état d'un Dispositif ou de contrôler son comportement à l'aide de MasterCommands (voir 7.3.7).

Les valeurs admissibles pour ces paramètres sont spécifiées dans le Tableau B.2.

Tableau B.2 – Types de MasterCommands

Valeur	MasterCommand	Description
0x00 à 0x59	Reserved	
0x5A	Repli	Transition du mode communication au mode SIO. Le Dispositif doit effectuer cette transition après 3 durées de MasterCycleTimes et avant que 500 ms ne se soient écoulées après la MasterCommand.
0x5B to 0x94	Réservé	
0x95	MasterIdent	Indique une version de Maître supérieure à 1.0
0x96	DeviceIdent	Début de la vérification de la page de Paramètres Directs pour ce qui concerne les entrées modifiées
0x97	DeviceStartup	Basculement du Dispositif de l'état OPERATE ou PREOPERATE à l'état STARTUP
0x98	ProcessDataOutputOperate	Données de Processus de sortie valides
0x99	DeviceOperate	Données de Processus de sortie non valides ou non disponibles. Basculement du Dispositif de l'état STARTUP ou PREOPERATE vers l'état OPERATE
0x9A	DevicePreoperate	Basculement du Dispositif de l'état STARTUP à l'état PREOPERATE
0x9B à 0xFF	Réservé	

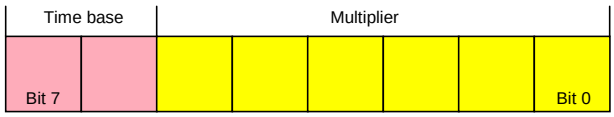
B.1.3 MasterCycleTime et MinCycleTime

Le MasterCycleTime est un paramètre du Maître qui établit la durée de cycle réelle d'un port particulier.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le MinCycleTime est un paramètre de Dispositif qui informe le Maître de la durée de cycle la plus courte prise en charge par ce Dispositif.

Pour l'application de MasterCycleTime et de MinCycleTime, voir A.3.7. La structure de ces deux paramètres est illustrée en Figure B.2.



Légende

Anglais	Français
Time base	Base de temps
Multiplier	Multiplicateur

Figure B.2 – MinCycleTime

Bits 0 à 5: Multiplicateur

Ces bits contiennent un multiplicateur sur 6 bits pour le calcul de MasterCycleTime ou de MinCycleTime. Les valeurs admissibles pour le multiplicateur sont 0 à 63.

Bits 6 à 7: Base de temps

Ces bits contiennent la base de temps pour le calcul de MasterCycleTime ou de MinCycleTime.

Lorsque tous les bits sont nuls, (code binaire 0x00), le Dispositif n'a aucun MinCycleTime. Dans ce cas, le Maître doit utiliser la temporisation de séquence M la plus défavorable calculée, c'est-à-dire avec le type de séquence M utilisé par le Dispositif, et les temps maximum pour t_A et t_2 (voir A.3.4 à A.3.6).

Les combinaisons admissibles de bases de temps et de multiplicateurs sont énumérées dans le Tableau B.3 avec les valeurs résultantes de MasterCycleTim ou de MinCycleTime.

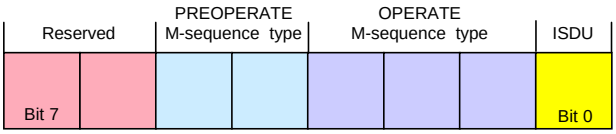
Tableau B.3 – Valeurs possibles de MasterCycleTime et de MinCycleTime

Codage de la base de temps	Valeur de la base de temps	Calcul	Temps de cycle
00	0,1 ms	Multiplicateur × Base de temps	0,4 ms à 6,3 ms
01	0,4 ms	6,4 ms + Multiplicateur × Base de temps	6,4 ms à 31,6 ms
10	1,6 ms	32,0 ms + Multiplicateur × Base de temps	32,0 ms à 132,8 ms
11	Réservé	Réservé	Réservé
NOTE La valeur 0,4 résulte du temps de transmission minimal potentiel selon A.3.7.			

B.1.4 Capacité de séquence M

La structure du paramètre de Capacité de séquence M est illustrée en Figure B.3.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Reserved	Réservé
PREOPERATE M-sequence type	Type de séquence M PREOPERATE
OPERATE M-sequence type	Type de séquence M OPERATE

Figure B.3 – Capacité de séquence M

Bit 0: ISDU

Ce bit indique la prise en charge ou non du canal de communication des ISDU. Les valeurs admissibles pour les ISDU sont données dans le Tableau B.4.

Tableau B.4 – Valeurs de ISDU

Valeur	Définition
0	ISDU non prises en charge
1	ISDU prises en charge

Bits 1 à 3: Codage de type de séquence M de l'état OPERATE

Ce paramètre indique le type de séquence M disponible pendant l'état OPERATE. Les codes admissibles pour le type de séquence M à l'état OPERATE sont énumérés dans le Tableau A.9 pour les Dispositifs patrimoniaux et dans le Tableau A.10 pour les Dispositifs conformes à la présente norme.

Bits 4 à 5: Codage de type de séquence M de l'état PREOPERATE

Ce paramètre indique le type de séquence M disponible pendant l'état PREOPERATE. Les codes admissibles pour le type de séquence M à l'état PREOPERATE sont énumérés dans le Tableau A.8.

Bits 6 à 7: Réservés

Ces bits sont réservés et doivent être remis à zéro dans la présente version de la spécification.

B.1.5 RevisionID (RID)

Le paramètre RevisionID est le numéro de version à deux chiffres du protocole SDCI actuellement utilisé dans le Dispositif. Sa structure est illustrée en Figure B.4. La valeur initiale de RevisionID à la mise sous tension est la valeur intrinsèque du RevisionID de protocole. Elle peut être écrasée (voir 10.6.3) jusqu'à la mise sous tension suivante.

La présente édition de la norme spécifie la version de protocole 1.1.

NOTE La version de protocole héritage 1.0 est spécifiée en [8].

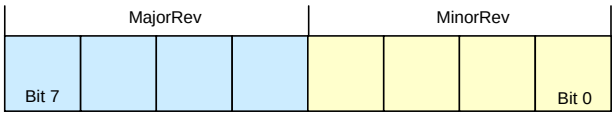


Figure B.4 – RevisionID

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Bits 0 à 3: MinorRev (révision mineure)

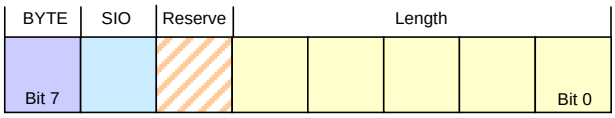
Ces bits contiennent les chiffres du numéro de version mineur, par exemple 0 pour la version de protocole 1.0. Les valeurs admissibles pour MinorRev sont 0x0 à 0xF.

Bits 4 à 7: MajorRev

Ces bits contiennent le chiffre principal du numéro de version, par exemple 1 pour la version de protocole 1.0. Les valeurs admissibles pour MajorRev sont 0x0 à 0xF.

B.1.6 ProcessDataIn

La structure du paramètre ProcessDataIn est illustrée en Figure B.5.



Légende

Anglais	Français
BYTE	OCTET (BYTE)
Reserve	Réservé
Length	Longueur

Figure B.5 – ProcessDataIn

Bits 0 à 4: Longueur

Ces bits contiennent la longueur des données d'entrée (Données de Processus du Dispositif vers le Maître) exprimée dans l'unité de longueur désignée par le bit du paramètre BYTE (octet). Les codes admissibles pour la longueur sont spécifiés dans le Tableau B.6.

Bit 5: Réservé

Ce bit est réservé et doit être mis à zéro dans la présente version de la spécification.

Bit 6: SIO

Ce bit indique si le Dispositif comporte un signal de commutation en mode SIO. Les valeurs admissibles pour le mode SIO sont énumérées dans le Tableau B.5.

Tableau B.5 – Valeurs de SIO

Valeur	Définition
0	Mode SIO non pris en charge
1	Mode SIO pris en charge

Bit 7: BYTE (OCTET)

Ce bit indique l'unité utilisée pour la "Longueur". Les valeurs admissibles de BYTE ainsi que la définition résultante de la longueur des Données de Processus correspondantes sont énumérées dans le Tableau B.6.

Tableau B.6 – Combinaisons admises de BYTE (octet) et Length (longueur)

OCTET (BYTE)	Longueur (Length)	Définition
0	0	Pas de Données de Processus
0	1	Données de Processus de 1 bit, structurées en bits
0	*n* (2 à 15)	Données de Processus de *n* bits, structurées en bits

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

OCTET (BYTE)	Longueur (Length)	Définition
0	16	Données de Processus de 16 bits, structurées en bits
0	17 à 31	Réservé
1	0, 1	Réservé
1	2	Données de Processus de 3 octets, structurées en octets
1	*n* (3 à 30)	Données de Processus de *n* +1 octets, structurées en octets
1	31	Données de Processus de 32 octets, structurées en octets

B.1.7 ProcessDataOut

La structure du paramètre ProcessDataOut est identique à celle de ProcessDataIn, à l'exception du bit 6 ("SIO") qui est réservé.

B.1.8 VendorID (VID)

Ces octets contiennent une valeur unique mondiale identifiant chaque fournisseur.

NOTE Les VendorID (identifiants de fournisseurs) sont attribués par le "IO-Link consortium".

B.1.9 DeviceID (DID)

Ces octets contiennent le DeviceID utilisé actuellement. Il n'est pas admis de valeur "0". La valeur initiale de DeviceID à la mise sous tension est sa valeur intrinsèque. Elle peut être écrasée (voir 10.6.2) jusqu'à la mise sous tension suivante.

NOTE Les paramètres de communication MinCycleTime, M-sequence Capability, Process Data In et Process Data Out peuvent être modifiés pour assurer la compatibilité au DeviceID demandé.

B.1.10 FunctionID (FID)

Ce paramètre sera défini dans une version ultérieure.

B.1.11 SystemCommand

Seuls les Dispositifs ne prenant pas en charge les ISDU doivent utiliser le paramètre SystemCommand dans la page 1 de Paramètres Directs. La mise en œuvre de SystemCommand est facultative. Pour une description détaillée des fonctions de SystemCommand, voir le Tableau B.9.

NOTE La SystemCommand sur la page 1 de Paramètres Directs ne fournit pas de réponse positive ou négative lors de l'exécution d'une fonction sélectionnée

B.1.12 Page 2 de Paramètres Directs spécifiques au Dispositif

Les Paramètres Directs spécifiques au Dispositif sont un jeu de paramètres disponibles pour la technologie spécifique au Dispositif. L'utilisation de Paramètres Directs spécifiques au Dispositif est facultative.

NOTE La liste complète des paramètres de la page 2 de Paramètres Directs est accessible en lecture ou en écriture via l'Index 1 (voir B.1.1).

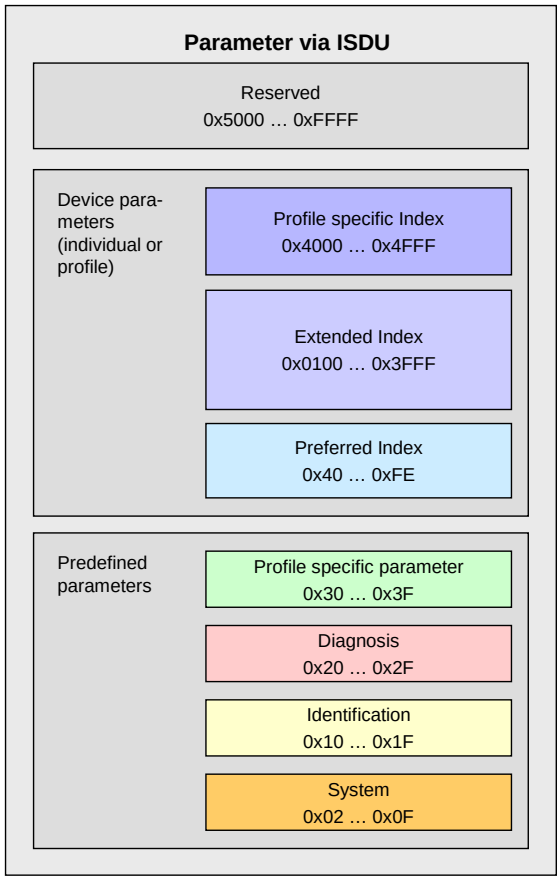
B.2 Paramètres de Dispositif prédéfinis

B.2.1 Aperçu

Les diverses et nombreuses technologies et conceptions de capteurs et d'actionneurs nécessitent un accès séparé et facile à des paramètres et à des commandes complexes bien

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

au-delà des capacités de la page 2 de Paramètres Directs. Du point de vue du Maître, ces paramètres et commandes complexes sont appelés objets de données d'application. Les éléments appelés "conteneurs" d'ISDU sont le moyen de transfert utilisé pour échanger des objets de données d'application ou des objets de données courts. L'Index des ISDU est utilisé pour solliciter les objets de données. La Figure B.6 illustre la mise en correspondance générale des objets de données pour la transmission des ISDU.



Légende

Anglais	Français
Parameter via ISDU	Paramètre via ISDU
Reserved	Réservé
0x5000 ... 0xFFFF	0x5000 à 0xFFFF
Device parameters (individual or profile)	Paramètres du Dispositif (individuel ou profil)
Profile specific Index	Index spécifique au profil
0x4000 ... 0x4FFF	0x4000 à 0x4FFF
Extended Index	Index étendu
0x0100 ... 0x3FFF	0x0100 à 0x3FFF
Preferred Index	Index préférentiel
0x40 ... 0xFE	0x40 à 0xFE
Predefined parameters	Paramètres prédéfinis
Profile specific parameter	Paramètre spécifique au profil
0x30 ... 0x3F	0x30 à 0x3F
Diagnosis	Diagnostic
0x20 ... 0x2F	0x20 à 0x2F
Identification	Identification

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
0x10 ... 0x1F	0x10 à 0x1F
System	Système
0x02 ... 0x0F	0x02 à 0x0F

Figure B.6 – Espace d'index pour des objets de données ISDU

L'Article B.2 contient des définitions et des exigences pour la mise en œuvre d'applications de Dispositifs spécifiques à la technologie. Les règles de mise en œuvre pour les paramètres et les commandes sont spécifiées dans le Tableau B.7.

Tableau B.7 – Règles de mise en œuvre des paramètres et commandes

Numéro de règle	Spécification de règle
1	Tous les paramètres d'un Index doivent pouvoir être lus et/ou écrits comme un objet de données entier, via le Sous-index 0
2	Lors du re-paramétrage, l'application de Dispositif spécifique à la technologie doit résoudre les incohérences de jeux de paramètres dépendants
3	La durée d'une demande de service ISDU est limitée (voir le Tableau 97). Une application du Maître peut abandonner des services ISDU après expiration de cette durée
4	Les commandes d'application (par exemple apprentissage, réinitialisation aux réglages d'usine, etc.) sont traitées comme des paramètres. L'exécution d'une commande d'application est confirmée par une réponse positive de service – Write.res(+). Une réponse négative de service – Write.res(-) – doit indiquer que l'exécution de la commande d'application a échoué. Dans les deux cas, les délais impartis doivent être pris en compte (voir le Tableau 97)

Le Tableau B.8 définit l'attribution des objets de données (paramètres et commandes) à la plage d'Index des ISDU. Tous les Index au-dessus de 2 concernent les ISDU.

Tableau B.8 – Attribution des Index d'objets de données (paramètres de Dispositif)

Index (déc)	Nom d'objet	Accès	Longueur	Type de Données	M/O/C	Remarque
0x0000 (0)	Page 1 de Paramètres Directs	R (lecture)		RecordT	M	Redirigé vers le canal de communication de page, voir 10.7.5
0x0001 (1)	Page 2 de Paramètres Directs	R/W (Lecture/Écriture)		RecordT	M	Redirigé vers le canal de communication de page, voir 10.7.5
0x0002 (2)	Commande système	W (écriture)	1 octet	UIntegerT	M/O	Définition du Code de Commande (Voir B.2.2)
0x0003 (3)	Index de Stockage des Données	R/W (Lecture/Écriture)	Variable	RecordT	M	Ensemble d'objets de données pour stockage (Voir B.2.3)
0x0004 à 0x000B (4 à 11)	Réservé					Réservé aux opérations exceptionnelles
0x000C (12)	Verrous d'accès au Dispositif	R/W (Lecture / Écriture)	2 octets	RecordT	C	Fonctions normalisées de verrouillage de Dispositifs (Voir B.2.4)
0x000D (13)	Caractéristique de profil	R (lecture)	Variable	ArrayT de UIntegerT16	O	Caractéristique de profil (voir B.2.5)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Index (déc)	Nom d'objet	Accès	Longueur	Type de Données	M/O/C	Remarque
0x000E (14)	Descripteur de PDInput	R (lecture)	Variable	ArrayT de OctetStringT3	O	Réservé au profil de Dispositif (voir B.2.6)
0x000F (15)	Descripteur de PDOutput	R (lecture)	Variable	ArrayT de OctetStringT3	O	Réservé au profil de Dispositif (voir B.2.7)
0x0010 (16)	Nom du fournisseur	R (lecture)	max. 64 octets	StringT	M	Informative (Voir B.2.8)
0x0011 (17)	Texte du fournisseur	R (lecture)	max. 64 octets	StringT	O	Informations supplémentaires sur le fournisseur (Voir B.2.9)
0x0012 (18)	Nom du produit	R (lecture)	max. 64 octets	StringT	M	Produit détaillé ou nom du type (Voir B.2.10)
0x0013 (19)	ID du produit	R (lecture)	max. 64 octets	StringT	O	Identification du produit ou du type (Voir B.2.11 pour plus de détails)
0x0014 (20)	Texte du produit	R (lecture)	max. 64 octets	StringT	O	Description de fonction ou de caractéristique de Dispositif (Voir B.2.12)
0x0015 (21)	Numéro de série	R (lecture)	max. 16 octets	StringT	O	Numéro de série spécifique au fournisseur (Voir B.2.13)
0x0016 (22)	Version du matériel	R (lecture)	max. 64 octets	StringT	O	Format spécifique au fournisseur (Voir B.2.14)
0x0017 (23)	Version du microprogramme	R (lecture)	max. 64 octets	StringT	O	Format spécifique au fournisseur (Voir B.2.15)
0x0018 (24)	Balise spécifique à l'application	R/W (Lecture/Écriture)	Min. 16, max. 32 octets	StringT	O	Emplacement ou fonction de balise, défini par l'utilisateur (Voir B.2.16)
0x0019 à 0x001F (25 à 31)	Réservé					
0x0020 (32)	Nombre d'erreurs	R (lecture)	2 octets	UIntegerT	O	Erreurs depuis la mise sous tension ou la réinitialisation (Voir B.2.17)
0x0021 à 0x0023 (33 à 35)	Réservé					
0x0024 (36)	État du Dispositif	R (lecture)	1 octet	UIntegerT	O	Contient l'état courant du Dispositif (Voir B.2.18)
0x0025 (37)	État détaillé du Dispositif	R (lecture)	Variable	ArrayT de OctetStringT3	O	voir B.2.19
0x0026 à 0x0027 (38 à 39)	Réservé					
0x0028 (40)	Process-DataInput	R (lecture)	Longueur de PD	Spécifique au Dispositif	O	Lecture des dernières Données de Processus valides depuis le canal PDin (Voir B.2.19)
0x0029 (41)	Process-DataOutput	R (lecture)	Longueur de PD	Spécifique au Dispositif	O	Lecture des dernières Données de Processus valides depuis le canal PDout (Voir B.2.21)
0x002 à 0x002F (42 à 47)	Réservé					

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Index (déc)	Nom d'objet	Accès	Longueur	Type de Données	M/O/C	Remarque
0x0030 (48)	Décalage temporel	R/W (Lecture / Écriture)	1 octet	RecordT	O	Synchronisation entre temps d'application du Dispositif et temps de séquence M (Voir B.2.22)
0x0031 à 0x003F (49 à 63)	Réservé aux profils					
0x0040 à 0x00FE (64 à 254)	Index préférentiel					Spécifique au Dispositif (8 bits)
0x00FF (255)	Réservé					
0x0100 à 0x3FFF (256 - 16383)	Index étendu					Spécifique au Dispositif (16 bits)
0x4000 à 0x4FFF (16384 - 20479)	Index spécifique au profil					Réservé au profil de Dispositif
0x5000 à 0xFFFF (20480 - 65535)	Réservé					
Légende	M = obligatoire; O = facultatif; C = conditionnel					

B.2.2 SystemCommand

Les Dispositifs prenant en charge des ISDU doivent utiliser l'Index ISDU 0x0002 pour recevoir la SystemCommand. Les commandes doivent être acquittées. Un acquittement positif indique la finalisation complète et correcte de la commande demandée. Un acquittement négatif indique que la commande ne peut être exécutée ou qu'elle s'est terminée par une erreur. Une SystemCommand doit être exécutée dans les 5 s afin de satisfaire aux exigences de temporisation des ISDU (voir le Tableau 97).

La mise en œuvre de la fonctionnalité SystemCommand est obligatoire pour le Maître et facultative pour le Dispositif. Le codage d'une SystemCommand est spécifié dans le Tableau B.9.

Tableau B.9 – Codage d'une SystemCommand (ISDU)

Commande (hex)	Commande (déc)	Nom de la commande	M/O	Définition
0x00	0	Réservé		
0x01	1	ParamUploadStart	O	Démarrage du téléchargement amont de paramètres
0x02	2	ParamUploadEnd	O	Arrêt du téléchargement amont de paramètres
0x03	3	ParamDownloadStart	O	Démarrage du téléchargement aval de paramètres
0x04	4	ParamDownloadEnd	O	Arrêt du téléchargement aval de paramètres

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Commande (hex)	Commande (déc)	Nom de la commande	M/O	Définition
0x05	5	ParamDownloadStore	O	Finalisation du paramétrage et début du Stockage des Données
0x06	6	ParamBreak	O	Annulation de toutes les commandes de paramétrage
0x07 à 0x3F	7 à 63	Réservé		
0x40 à 0x7F	64 à 127	Réservé aux profils		
0x80	128	Réinitialisation du Dispositif	O	
0x81	129	Réinitialisation de l'application	O	
0x82	130	Restauration des réglages d'usine	O	
0x83 à 0x9F	131 à 159	Réservé		
0xA0 à 0xFF	160 à 255	Spécifique au fournisseur		
NOTE Voir 10.3 **Légende** M = obligatoire; O = facultatif				

La SystemCommand 0x05 (ParamDownloadStore) doit être utilisée conformément au 10.4.2 lorsque le Dispositif fournit des paramètres à enregistrer via le mécanisme de Stockage de Données; en d'autres termes, le paramètre "Index_List" dans l'Index 0x0003 n'est pas vide (voir le Tableau B.10).

La mise en œuvre des SystemCommands 0x01 à 0x06 exigées pour le paramétrage de bloc conformément au 10.3.5 est facultative. Cependant, on doit utiliser toutes ces commandes ou aucune d'entre elles (pour la SystemCommand 0x05, la règle applicable au Stockage des Données prévaut).

Pour les options de SystemCommand sur la page 1 de Paramètres Directs, voir B.1.11.

B.2.3 Index de Stockage de Données

Le Tableau B.10 spécifie les attributions d'Index de Stockage de Données.

Tableau B.10 – Attributions d'Index de Stockage de Données

Index	Sous-index	Accès	Nom du paramètre	Codage	Type de Données
0x0003	01	R/W (Lecture/Écriture)	DS_Command	0x00: Réservé 0x01: DS_UploadStart 0x02: DS_UploadEnd 0x03: DS_DownloadStart 0x04: DS_DownloadEnd 0x05: DS_Break 0x06 à 0xFF: Réservé	UintegerT8 (8 bits)

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Index	Sous-index	Accès	Nom du paramètre	Codage	Type de Données
	02	R	State_Property	Bit 0: Réservé Bits 1 et 2: État du Stockage des Données 0b00: Inactif 0b01: Téléchargement amont 0b10: Téléchargement aval 0b11: Stockage des Données verrouillé Bits 3 à 6: Réservé Bit 7: Drapeau DS_UPLOAD_FLAG "1": Demande DS_UPLOAD_REQ en cours "0": Pas de demande DS_UPLOAD_REQ	UintegerT8 (8 bits)
	03	R	Data_Storage_ Size	Nombre d'octets pour le stockage de toutes les informations nécessaires au remplacement du Dispositif (voir 10.4.5). La taille maximale est de 2 048 octets.	UintegerT16 (32 bits)
	04	R	Parameter_ Checksum	Indication d'une version de jeu de paramètres: Signature CRC ou compteur de versions (voir 10.4.8)	UintegerT32 (32 bits)
	05	R	Index_List	Liste des index de paramètres à sauvegarder (voir le Tableau B.11)	OctetStringT (variable)

L'Index de Stockage des Données de paramètres 0x0003 contient toutes les informations à utiliser pour le traitement du Stockage de Données. Ce paramètre est réservé pour des échanges privés entre le Maître et le Dispositif; le Maître doit bloquer toute demande d'accès provenant d'une application passerelle vers cet Index (voir la Figure 4). Les paramètres dans cet Index 0x0003 sont les suivants.

DS_Command

Cet octet contient les commandes de Stockage des Données pour le Dispositif.

State_Property

Cet octet indique la valeur courante du mécanisme de Stockage de Données. Le bit 7 doit être enregistré en mémoire non volatile. Le Maître vérifie ce bit au démarrage et effectue si nécessaire un téléchargement amont des paramètres.

Data_Storage_Size

Ces quatre octets indiquent la taille de mémoire requise en nombre d'octets, pour stocker toutes les informations nécessaires au remplacement d'un Dispositif, y compris les informations de structure (Index, Sous-index). Le type de données est UintegerT32 (32 bits). La taille maximale est de 2 048 octets. Voir le Tableau F.1 pour les éléments à prendre en compte dans le calcul de la taille.

Parameter_Checksum

Cette somme de contrôle est utilisée pour détecter les modifications apportées au jeu de paramètres sans lire tous les paramètres. La valeur de la somme de contrôle est calculée selon la procédure décrite en 10.4.8. Le Dispositif doit modifier la somme de contrôle lorsqu'un paramètre du jeu de paramètres a été modifié. Des jeux de paramètres différents doivent avoir des sommes de contrôle différentes. Il est recommandé que le Dispositif conserve ces paramètres localement dans une mémoire non volatile.

Index_List

Le Tableau B.11 représente la structure de l'Index_List. Chaque Index_List peut comporter 70 entrées au maximum (voir le Tableau 97).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau B.11 – Structure de Index_List

Entrée	Adresse	Définition	Type de Données
X1	Index	Index du premier paramètre à sauvegarder	Unsigned16
	Sous-index	Sous-index du premier paramètre à sauvegarder	Unsigned8
X2	Index	Index du paramètre suivant à sauvegarder	Unsigned16
	Sous-index	Sous-index du paramètre suivant à sauvegarder	Unsigned8
…..	……	……	…….
Xn	Index	Index du dernier paramètre à sauvegarder	Unsigned16
	Sous-index	Sous-index du dernier paramètre à sauvegarder	Unsigned8
Xn+1	Index	Marqueur de fin 0x0000: Fin de la Index_List >0x0000: Index suivant contenant une Index_List	Unsigned16

Les grands jeux de paramètres peuvent être traités via des Index_Lists concaténées. Les deux derniers octets de l'Index_List doivent détenir le Marqueur de Fin. Une valeur "0" indique la fin de la Liste d'Index. En cas de concaténation, le Marqueur de Fin est placé dans l'Index suivant contenant une Liste d'Index. La structure de la Liste d'Index suivante est identique à celle spécifiée dans le Tableau B.11. Ainsi, la concaténation des listes s'achève si un Marqueur de Fin de valeur "0" est trouvé.

B.2.4 Verrous d'accès au Dispositif

Le paramètre Verrous d'accès au Dispositif permet de contrôler le comportement du Dispositif. Les fonctions de Dispositif normalisées peuvent être configurées de manière indépendante au moyen de drapeaux définis dans ce paramètre. La configuration des verrous d'accès au Dispositif peut être modifiée en écrasant le paramètre correspondant. La configuration proprement dite est effectuée par accès en lecture à ce paramètre. Le type de données est RecordT de BooleanT. L'accès est uniquement autorisé via le Sous-index 0. Ce paramètre est facultatif. S'il est utilisé, il doit être non volatil.

Les catégories suivantes de verrous d'accès au Dispositif sont spécifiées:

- Accès en écriture aux paramètres (facultatif);
- Stockage des Données (obligatoire si le Dispositif prend en charge le Stockage de Données);
- Paramétrage local (facultatif);
- Fonctionnement de l'interface utilisateur local (facultatif).

Le Tableau B.12 énumère les possibilités de verrouillage du Dispositif.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau B.12 – Possibilités de verrouillage du Dispositif

Bit	Catégorie	Définition
0	Accès (en écriture) aux paramètres (facultatif)	0: déverrouillé (par défaut) 1: verrouillé
1	Stockage de données (obligatoire si le Dispositif prend en charge le stockage de données)	0: déverrouillé (par défaut) 1: verrouillé (voir la NOTE)
2	Paramétrage local (facultatif)	0: déverrouillé (par défaut) 1: verrouillé
3	Interface utilisateur local (facultatif)	0: déverrouillé (par défaut) 1: verrouillé
4 – 15	Réservé	
NOTE Le Maître lit les paramètres State_Property/State du Stockage de données des paramètres (voir le Tableau B.10) avant toute action.		

Accès (en écriture) aux paramètres

Si ce bit est établi, l'accès en écriture à tous les paramètres du Dispositif sur l'interface de communication SDCI est interdit pour tous les paramètres de lecture/écriture du Dispositif, à l'exception du paramètre Verrou d'accès au Dispositif. L'accès en lecture n'est pas concerné. Le Dispositif doit répondre par une réponse négative de service – accès refusé – à un accès en écriture si l'accès au paramètre est verrouillé.

Le mécanisme du verrou d'accès (en écriture) au paramètre ne doit pas bloquer les opérations de téléchargement aval du mécanisme de Stockage des Données (entre DS_DownloadStart et DS_DownloadEnd ou DS_Break).

Stockage de Données

Si ce bit est établi dans le Dispositif, le mécanisme de Stockage des Données est désactivé (voir 10.4.2 et 11.3.3). Dans ce cas, le Dispositif doit répondre à un accès en écriture (dans son Index de Stockage de Données) par une réponse négative de service – accès refusé – (voir B.2.3). L'accès en lecture à l'Index de Stockage des Données n'est pas concerné.

Cette configuration est également indiquée dans la propriété d'état au sein de l'Index de Stockage de Données.

Paramétrage local

Si ce bit est établi, le paramétrage au moyen d'éléments de commande locaux sur le Dispositif est neutralisé.

Interface utilisateur local

Si ce bit est établi, le fonctionnement de l'interface homme-machine sur le Dispositif est désactivé.

B.2.5 Caractéristique de profil

Ce paramètre contient la liste des ProfileIdentifiers (PID) correspondant au Profil de Dispositif mis en œuvre dans le Dispositif.

NOTE Les informations détaillées sont fournies dans la référence [7].

B.2.6 Descripteur d'entrée de PD

Ce paramètre contient la description de la structure des données de processus d'entrée pour un profil de Dispositif.

NOTE Les informations détaillées sont fournies dans la référence [7].

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.7 Descripteur de sortie de PD

Ce paramètre contient la description de la structure des données de processus de sortie pour un profil de Dispositif.

NOTE Les informations détaillées sont fournies dans la référence [7].

B.2.8 Nom du fournisseur

Le paramètre Nom du Fournisseur contient uniquement l'un des noms de fournisseurs énumérés pour le VendorID attribué. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength (longueur fixe) maximale de 64. Ce paramètre est obligatoire.

NOTE La liste des noms de fournisseur associés à un VendorID donné est conservée par le IO-Link consortium.

B.2.9 Texte du fournisseur

Le paramètre Texte du fournisseur contient des informations supplémentaires sur le fournisseur. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est facultatif.

B.2.10 Nom du produit

Le paramètre Nom de produit contient le nom de produit complet. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est obligatoire.

NOTE L'entrée correspondante dans la liste des variantes de Dispositif de l'IODD est supposée correspondre à ce paramètre.

B.2.11 ID du produit

Le paramètre ID du Produit doit contenir le produit spécifique au fournisseur ou l'identification du type de Dispositif. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est facultatif.

B.2.12 Texte de produit

Le paramètre Texte de produit doit contenir des informations de produit supplémentaires concernant le Dispositif, telles que la catégorie de produit (par exemple, capteur photoélectrique d'élimination de bruit de fond, capteur ultrasonore de la distance, capteur de la pression, etc.). Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est facultatif.

B.2.13 SerialNumber

Le paramètre SerialNumber doit contenir un code unique spécifique au fournisseur pour chaque Dispositif particulier. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 16. Ce paramètre est facultatif.

B.2.14 Version du matériel

Le paramètre Version du matériel doit contenir un code spécifique au fournisseur pour la version matérielle du Dispositif. Le paramètre est un objet de données en lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est facultatif.

B.2.15 Version du microprogramme

Le paramètre Version du microprogramme doit contenir un code spécifique au fournisseur pour la version de microprogramme du Dispositif. Le paramètre est un objet de données en

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

lecture seule. Le type de données est StringT avec une fixedLength maximale de 64. Ce paramètre est facultatif.

B.2.16 Balise spécifique à l'application

Le paramètre "Application Specific Tag" doit être prévu comme un objet de données de lecture/écriture pour l'application utilisateur. Il peut servir de "fonction de balise" (rôle du Dispositif) ou d'"emplacement de balise" (emplacement du Dispositif). Le type de données est StringT avec une fixedLength minimale de 16 et une fixedLength maximale de 32. Par défaut, il est recommandé de remplir ce paramètre avec "***". Ce paramètre est facultatif.

NOTE Dans le domaine de l'automatisation de processus, cette longueur est en général de 32 octets.

B.2.17 Nombre d'erreurs

Le paramètre Nombre d'erreurs fournit des informations sur les erreurs apparues dans l'application du Dispositif depuis la mise sous tension ou la réinitialisation. Son utilisation est spécifique au fournisseur ou au Dispositif. Le type de données est UIntegerT avec une bitLength de 16. Le paramètre est un objet de données en lecture seule. Ce paramètre est facultatif.

B.2.18 État du Dispositif

B.2.18.1 Aperçu

Le paramètre État du Dispositif doit fournir des informations sur l'était du Dispositif (diagnostic) sur la base de la technologie du Dispositif. Le type de données est UIntegerT avec une bitLength de 8. Le paramètre est un objet de données en lecture seule. Ce paramètre est facultatif.

Les états de Dispositif sont spécifiés dans le Tableau B.13 ci-dessous. Ils doivent être générés par les applications du Dispositif. Le paramètre État du Dispositif peut être examiné par tout programme d'Automate programmable ou par des outils tels que la Gestion des Actifs (voir Article 11).

Le Tableau B.13 énumère les différentes informations d'État du Dispositif. Les critères applicables à ces indications sont spécifiés de B.2.18.2 à B.2.18.5.

Tableau B.13 – Paramètre d'état du Dispositif

Valeur	Définition
0	Le Dispositif fonctionne correctement
1	Maintenance requise (voir B.2.18.2)
2	Hors spécifications (voir B.2.18.3)
3	Vérification fonctionnelle (voir B.2.18.4)
4	Défaillance (voir B.2.18.5)
5 – 255	Réservé

B.2.18.2 Maintenance requise

Bien que les Données de Processus soient valides, les diagnostics internes indiquent que le Dispositif est sur le point de perdre son aptitude à fonctionner correctement.

EXEMPLES Lentilles optiques sales, accumulation de dépôts, niveau de lubrifiant bas.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.18.3 Hors spécifications

Bien que les Données de Processus soient valides, les diagnostics internes indiquent que le Dispositif fonctionne hors de la plage de mesure spécifiée ou des conditions environnementales spécifiées.

EXEMPLES Tension d'alimentation, énergie auxiliaire, température, pression pneumatique, perturbations électromagnétiques, vibrations, accélération, interférence de l'éclairage, formation de bulles dans les liquides.

B.2.18.4 Vérification fonctionnelle

Les Données de Processus sont temporairement non valides du fait de manipulations volontaires sur le Dispositif.

EXEMPLES Etalonnages, apprentissage, réglages de position, simulation.

B.2.18.5 Défaillance

Les Données de Processus sont non valides du fait d'un dysfonctionnement du Dispositif ou de ses périphériques. Le Dispositif est incapable de remplir sa fonction prévue.

B.2.19 État détaillé du Dispositif

Le paramètre État détaillé du Dispositif doit fournir des informations concernant les Événements actuellement en attente dans le Dispositif. Les Événements de TYPE "Erreur" ou "Avertissement" et de MODE "Apparition d'Événement" (voir A.6.4) doivent être saisis dans la liste d'États détaillés du Dispositif avec les EventQualifier et EventCode. Lorsqu'il y a un Événement de MODE "Disparition d'Événement", l'entrée correspondante de EventQualifier, dans l'État détaillé du Dispositif, doit être mise à "0x00" et le EventCode à "0x0000". De cette manière, ce paramètre fournit toujours l'état de diagnostic courant du Dispositif. Le paramètre est un objet de données en lecture seule. Le Type de Données est ArrayT avec un nombre maximal de 64 éléments de matrice (entrées d'Événements). Le nombre d'éléments de matrice de ce paramètre est spécifique au Dispositif. Lors de la mise hors tension ou de la réinitialisation du Dispositif, le contenu de tous les éléments de la matrice est remis aux valeurs initiales – EventQualifier "0x00", EventCode "0x0000". Ce paramètre est facultatif.

Le Tableau B.14 spécifie la structure du paramètre État détaillé du Dispositif.

Tableau B.14 – État détaillé du Dispositif (Index 0x0025)

Sous-index	Nom d'objet	Type de données	Commentaire
1	Error_Warning_1	3 octets	Tous les octets 0x00: Pas d'erreur/ Avertissement Octet 1: EventQualifier Octet 2,3: EventCode
2	Error_Warning_2	3 octets	
3	Error_Warning_3	3 octets	
4	Error_Warning_4	3 octets	
…			
n	Error_Warning_n	3 octets	

Le concepteur peut choisir d'utiliser une liste statique, c'est-à-dire une position de matrice fixe pour chaque Événement, avec un EventCode spécifique, ou une liste dynamique, c'est-à-dire que chaque entrée d'Événement est enregistrée dans la position libre suivante de la matrice. L'accès aux sous-index n'est pas admis pour une liste dynamique.

B.2.20 ProcessDataInput

Le paramètre ProcessDataInput doit fournir les dernières données d'entrée de processus valides provenant de l'application du Dispositif. Le type et la structure des données sont

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

identiques aux Données de Processus d'entrée transférées sur le canal de communication de processus. Le paramètre est un objet de données en lecture seule. Ce paramètre est facultatif.

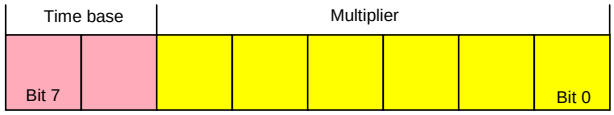
B.2.21 ProcessDataOutput

Le paramètre ProcessDataOutput doit fournir les dernières données de sortie de processus valides destinées à l'application du Dispositif. Le type et la structure des données sont identiques aux Données de Processus de sortie transférées sur le canal de communication de processus. Le paramètre est un objet de données en lecture seule. Ce paramètre est facultatif.

B.2.22 Décalage temporel

Le paramètre Décalage Temporel permet à une application du Dispositif de se synchroniser sur des cycles de séquence M de la couche de liaison de données grâce à des décalages temporels réglables. Le type de données est RecordT. L'accès est uniquement autorisé via le Sous-index 0. Le paramètre est un objet de données en lecture/écriture. Ce paramètre est facultatif.

La structure du paramètre Décalage temporel est illustrée en Figure B.7:



Légende

Anglais	Français
Time base	Base de temps
Multiplier	Multiplicateur

Figure B.7 – Structure du Décalage temporel

Bits 0 à 5: Multiplicateur

Ces bits contiennent un facteur sur 6 bits permettant de calculer le Décalage temporel. Les valeurs admissibles pour le multiplicateur sont 0 à 63.

Bits 6 à 7: Base de temps

Ces bits contiennent la base de temps utilisée pour le calcul du Décalage temporel.

Les combinaisons admissibles de la base de temps et du multiplicateur sont énumérées dans le Tableau B.15, ainsi que les valeurs résultantes du Décalage temporel. La remise à zéro du Multiplicateur et de la Base de Temps désactive la synchronisation au moyen d'un Décalage temporel. La valeur du Décalage temporel ne doit pas dépasser le temps MasterCycleTime (voir B.1.3)

Tableau B.15 – Codage de la base de temps et valeurs du décalage temporel

Codage de la base de temps	Valeur de la base de temps	Calcul	Décalage temporel
00	0,01 ms	Multiplicateur × Base de temps	0,01 ms à 0,63 ms
01	0,04 ms	0,64 ms + Multiplicateur × Base de temps	0,64 ms à 3,16 ms
10	0,64 ms	3,20 ms + Multiplicateur × Base de temps	3,20 ms à 43,52 ms
11	2,56 ms	44,16 ms + Multiplicateur × Base de temps	44,16 ms à 126,08 ms

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

B.2.23 Paramètre de profil (réservé)

Les index 0x0031 à 0x003F sont réservés aux profils de Dispositifs.

NOTE Les informations détaillées sont fournies dans la référence [7].

B.2.24 Index préférentiel

Des index préférentiels (0x0040 à 0x00FE) peuvent être utilisés pour des fonctions de Dispositifs spécifiques au fournisseur. Cette plage d'index est considérée préférable du fait de la surcharge système moindre due au protocole dans les ISDU et ainsi un débit de données supérieur pour de petits objets de données en comparaison à l'Index étendu (voir B.2.25).

B.2.25 Index étendu

Des index étendus (0x0100 à 0x3FFF) peuvent être utilisés pour des fonctions de Dispositifs spécifiques au fournisseur.

B.2.26 Index spécifique au profil (réservé)

Les index 0x4000 à 0x4FFF sont réservés aux profils de Dispositifs.

NOTE Les informations détaillées sont fournies dans la référence [7].

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe C
(normative)

ErrorTypes (erreurs d'ISDU)

C.1 Généralités

Un ErrorType est utilisé dans des confirmations de service négatives d'ISDU (voir A.5.2 et le Tableau A.13). Il indique la cause d'une confirmation négative d'un service de Lecture ou d'Écriture. L'origine de l'erreur peut se trouver dans le Maître (local) ou dans le Dispositif (distant).

L'ErrorType est constitué de deux octets, de la cause d'erreur principale et d'informations plus spécifiques:

- ErrorCode (octet de poids fort);
- AdditionalCode (octet de poids faible).

ErrorType fournit les informations concernant l'incident, l'origine et l'instance. Les ErrorTypes admissibles et les critères pour leur déploiement sont donnés en C.2 et C.3. Toutes les autres valeurs d'ErrorType sont réservées et ne doivent pas être utilisées.

C.2 ErrorTypes relatifs à l'application

C.2.1 Aperçu

Les ErrorTypes admissibles provenant de l'application du Dispositif sont énumérés dans le Tableau C.1.

Tableau C.1 – ErrorTypes

Incident	Code d'erreur	Code supplémentaire	Nom	Définition
Erreur d'application du Dispositif – pas de détails	0x80	0x00	APP_DEV	Voir C.2.2
Index non disponible	0x80	0x11	IDX_NOTAVAIL	Voir C.2.3
Sous-index non disponible	0x80	0x12	SUBIDX_NOTAVAIL	Voir C.2.4
Service temporairement non disponible	0x80	0x20	SERV_NOTAVAIL	Voir C.2.5
Service temporairement non disponible – contrôle local	0x80	0x21	SERV_NOTAVAIL_LOCCTRL	Voir C.2.6
Service temporairement non disponible – contrôle du Dispositif	0x80	0x22	SERV_NOTAVAIL_DEVCTRL	Voir C.2.7
Accès refusé	0x80	0x23	IDX_NOT_WRITEABLE	Voir C.2.8
Valeur de paramètre hors limites	0x80	0x30	PAR_VALOUTOFRNG	Voir C.2.9
Valeur de paramètre supérieure à la limite	0x80	0x31	PAR_VALGTLIM	Voir C.2.10
Valeur de paramètre inférieure à la limite	0x80	0x32	PAR_VALLTLIM	Voir C.2.11
Dépassement de	0x80	0x33	VAL_LENOVRRUN	Voir C.2.12

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Incident	Code d'erreur	Code supplémentaire	Nom	Définition
longueur de paramètre				
Longueur de paramètre insuffisante	0x80	0x34	VAL_LENUNDRUN	Voir C.2.13
Fonction non disponible	0x80	0x35	FUNC_NOTAVAIL	Voir C.2.14
Fonction temporairement indisponible	0x80	0x36	FUNC_UNAVAILTEMP	Voir C.2.15
Jeu de paramètres non valide	0x80	0x40	PAR_SETINVALID	Voir C.2.16
Jeu de paramètres incohérent	0x80	0x41	PAR_SETINCONSIST	Voir C.2.17
Application non prête	0x80	0x82	APP_DEVNOTRDY	Voir C.2.18
Spécifique au fournisseur	0x81	0x00	UNSPECIFIC	Voir C.2.19
Spécifique au fournisseur	0x81	0x01 à 0xFF	VENDOR_SPECIFIC	Voir C.2.19

C.2.2 Erreur d'application du Dispositif – pas de détails

Ce ErrorType doit être utilisé si le service demandé a été refusé par l'application du Dispositif et qu'aucune information détaillée de l'incident n'est disponible.

C.2.3 Index non disponible

Ce ErrorType doit être utilisé lorsqu'un accès en lecture ou en écriture concerne un Index inexistant.

C.2.4 Sous-index non disponible

Ce ErrorType doit être utilisé lorsqu'un accès en lecture ou en écriture concerne un Sous-index inexistant.

C.2.5 Service temporairement non disponible

Ce ErrorType doit être utilisé si un paramètre n'est pas accessible à un service en lecture ou en écriture, du fait de l'état courant de l'application du Dispositif.

C.2.6 Service temporairement non disponible – contrôle local

Ce ErrorType doit être utilisé si un paramètre n'est pas accessible à un service en lecture ou en écriture du fait d'une opération locale en cours au niveau du Dispositif (par exemple utilisation ou paramétrage au moyen d'un tableau de commande de Dispositif embarqué).

C.2.7 Service temporairement non disponible – contrôle du Dispositif

Ce ErrorType doit être utilisé si un service en lecture ou en écriture n'est pas accessible du fait d'un état déclenché à distance de l'application du Dispositif (par exemple le paramétrage au cours d'une opération d'apprentissage ou d'étalonnage déclenchée à distance).

C.2.8 Accès refusé

Ce ErrorType doit être utilisé si un service tente d'accéder en écriture à un paramètre en lecture seule.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.2.9 Valeur de paramètre hors limites

Ce ErrorType doit être utilisé pour un service en écriture d'un paramètre hors de sa plage de valeurs autorisée.

C.2.10 Valeur de paramètre supérieure à la limite

Ce ErrorType doit être utilisé pour un service en écriture d'un paramètre au-dessus de sa plage de valeurs spécifiée.

C.2.11 Valeur de paramètre inférieure à la limite

Ce ErrorType doit être utilisé pour un service en écriture d'un paramètre en dessous de sa plage de valeurs spécifiée.

C.2.12 Dépassement de longueur de paramètre

Ce ErrorType doit être utilisé lorsque le contenu d'un service en écriture d'un paramètre est plus important que la longueur spécifiée de ce même paramètre. Ce ErrorType doit également être utilisé si un objet de données est trop important pour être traité par l'application du Dispositif (par exemple, restriction de capacité du tampon d'ISDU).

C.2.13 Longueur de paramètre insuffisante

Ce ErrorType doit être utilisé lorsque le contenu d'un service en écriture d'un paramètre est moins important que la longueur spécifiée de ce même paramètre (par exemple accès en écriture d'une valeur Unsigned16 vers un paramètre Unsigned32).

C.2.14 Fonction non disponible

Ce ErrorType doit être utilisé pour un service en écriture ayant une valeur de commande non prise en charge par l'application du Dispositif (par exemple une SystemCommand dont la valeur n'est pas utilisée).

C.2.15 Fonction temporairement indisponible

Ce ErrorType doit être utilisé pour un service en écriture ayant une valeur de commande qui invoque une fonction de Dispositif non disponible, du fait de l'état courant de l'application du Dispositif (par exemple une SystemCommand).

C.2.16 Jeu de paramètres non valide

Ce ErrorType doit être utilisé si des valeurs transférées via un paramètre unique sont en conflit avec d'autres réglages de paramètres réels (par exemple points de consigne se chevauchant pour un réglage de données binaires – voir 10.3.4).

C.2.17 Jeu de paramètres incohérent

Ce ErrorType doit être utilisé à la fin d'un transfert de paramètre de bloc avec ParamDownloadEnd ou ParamDownloadStore si le contrôle de vraisemblance présente des incohérences (voir 10.3.5 et B.2.2).

C.2.18 Application non prête

Ce ErrorType doit être utilisé si un service en lecture ou en écriture est refusé du fait d'une application temporairement indisponible (par exemple les contrôleurs périphériques pendant le démarrage).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.2.19 Spécifique au fournisseur

Ce ErrorType doit être directement propagé aux éléments de traitement de niveau supérieur en tant qu'erreur (et non avertissement) par le Maître.

C.3 ErrorTypes dérivés

C.3.1 Aperçu

Les ErrorTypes dérivés sont générés dans la couche AL du Maître et sont dus à des incidents internes ou reçus du Dispositif. Le Tableau C.2 énumère les ErrorTypes dérivés spécifiés.

Tableau C.2 – ErrorTypes dérivés

Incident	Code d'erreur	Code supplémentaire	Nom	Définition
Maître – Erreur de communication	0x10	0x00	COM_ERR	Voir C.3.2
Maître – Expiration du délai d'ISDU	0x11	0x00	I-SERVICE_TIMEOUT	Voir C.3.3
Événement de Dispositif – Erreur d'ISDU (DL, Erreur, occurrence unique, 0x5600)	0x11	0x00	I-SERVICE_TIMEOUT	Voir C.3.4
Événement de Dispositif – Primitive de service ISDU illicite (AL, Erreur, occurrence unique, 0x5800)	0x11	0x00	I-SERVICE_TIMEOUT	Voir C.3.5
Maître – Erreur de somme de contrôle d'ISDU	0x56	0x00	M_ ISDU_CHECKSUM	Voir C.3.6
Maître – Primitive de service ISDU illicite	0x57	0x00	M_ ISDU_ILLEGAL	Voir C.3.7
Événement de Dispositif – Débordement de tampon d'ISDU (DL, Erreur, occurrence unique, 0x5200)	0x80	0x33	VAL_LENOVRRUN	Voir C.3.8 et C.2.12 Les Événements provenant de Dispositifs patrimoniaux doivent être redirigés en mode de compatibilité vers ce ErrorType dérivé.

C.3.2 Maître – Erreur de communication

Le Maître génère une réponse de service négative avec ce ErrorType si une erreur de communication a eu lieu pendant un service de lecture ou d'écriture, par exemple lorsque la connexion SDCI est interrompue.

C.3.3 Maître – Expiration du délai d'ISDU

Le Maître génère une réponse de service négative avec ce ErrorType, si un service de lecture ou d'écriture demeure en attente au-delà du délai de "Service I" spécifié (voir le Tableau 97) dans le Maître.

C.3.4 Événement de Dispositif – Erreur d'ISDU

Si le Maître a reçu un Événement avec le EventQualifier (voir A.6.4: DL, Erreur, occurrence unique d'Événement) et le EventCode 0x5600, il est généré une réponse de service négative indiquant une expiration du délai de service; cette réponse est renvoyée au demandeur (voir C.3.3).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

C.3.5 Événement de Dispositif – Primitive de service ISDU illicite

Si le Maître a reçu un Événement avec le EventQualifier (voir A.6.4: AL, Erreur, occurrence unique d'Événement) et le EventCode 0x5800, il est généré une réponse de service négative indiquant une expiration du délai de service; cette réponse est renvoyée au demandeur (voir C.3.3).

C.3.6 Maître – Erreur de somme de contrôle d'ISDU

Le Maître génère une réponse de service négative avec ce ErrorType, si sa Couche de Liaison de Données détecte une erreur de somme de contrôle de ses ISDU.

C.3.7 Maître – Primitive de service ISDU illicite

Le Maître génère une réponse de service négative avec ce ErrorType, si sa Couche de Liaison de Données détecte une primitive de service ISDU illicite.

C.3.8 Événement de Dispositif – Débordement de tampon d'ISDU

Si le Maître a reçu un Événement avec le EventQualifier (voir A.6.4: DL, Erreur, occurrence unique d'Événement) et le EventCode 0x5200, il est généré une réponse de service négative indiquant un dépassement de la longueur du paramètre; cette réponse est renvoyée au demandeur (voir C.2.12).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe D
(normative)

EventCodes (informations de diagnostic)

D.1 Généralités

Le concept d'Événements est décrit en 7.3.8.1; la structure générale et le codage des Événements sont spécifiés en A.6. Lorsque le StatusCode indique un Événement en cas d'incident de Dispositif ou de Maître, le EventCode correspondant doit être fourni en tant qu'informations de diagnostic. Comme spécifié en A.6, l'entrée "Événement" contient, outre le EventQualifier, un EventCode. Le EventCode identifie un incident réel. Les valeurs admissibles de "EventCode" sont énumérées dans le Tableau D.1. Toutes les autres valeurs d'EventCode sont réservées et ne doivent pas être utilisées.

D.2 EventCodes pour des Dispositifs

Le Tableau D.1 énumère les identifiants de EventCode spécifiés; ainsi que leurs définitions. Les EventCodes sont générés par l'application spécifique à la technologie du Dispositif (instance = APP).

Tableau D.1 – EventCodes

EventCodes	Définition et action de maintenance recommandée	Valeur d'état de Dispositif (NOTE 1)	TYPE (NOTE 2)
0x0000	Pas de dysfonctionnement	0	Notification
0x1000	Dysfonctionnement général – erreur inconnue	4	Erreur
0x1001 à 0x17FF	Réservé		
0x1800 à 0x18FF	Spécifique au fournisseur		
0x1900 à 0x3FFF	Réservé		
0x4000	Défaut de température – Surcharge	4	Erreur
0x4001 à 0x420F	Réservé		
0x4210	Dépassement de température du Dispositif – Éliminer la source de chaleur	2	Avertissement
0x4211 à 0x421F	Réservé		
0x4220	Température de dispositif insuffisante– Isoler le Dispositif	2	Avertissement
0x4221 à 0x4FFF	Réservé		
0x5000	Défaut matériel du Dispositif – Remplacer le Dispositif	4	Erreur
0x5001 à 0x500F	Réservé		
0x5010	Dysfonctionnement d'un composant– Réparer ou remplacer	4	Erreur
0x5011	Perte de mémoire non volatile – Vérifier les batteries	4	Erreur
0x5012	Batteries faibles – Remplacer les batteries	2	Avertissement
0x5013 à 0x50FF	Réservé		
0x5100	Défaut général d'alimentation – Vérifier la disponibilité de l'alimentation	4	Erreur
0x5101	Fusible fondu/ouvert – Remplacer le fusible	4	Erreur
0x5102 à 0x510F	Réservé		

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

EventCodes	Définition et action de maintenance recommandée	Valeur d'état de Dispositif (NOTE 1)	TYPE (NOTE 2)
0x5110	Surtension de l'alimentation principale – Vérifier la marge de tolérance	2	Avertissement
0x5111	Sous-tension de l'alimentation principale – Vérifier la marge de tolérance	2	Avertissement
0x5112	Défaut de la tension d'alimentation secondaire (classe de port B) – Vérifier la marge de tolérance	2	Avertissement
0x5113 à 0x5FFF	Réservé		
0x6000	Défaut logiciel du Dispositif – Vérifier la version du microprogramme	4	Erreur
0x6001 à 0x631F	Réservé		
0x6320	Erreur de paramètre – Vérifier la fiche technique et les valeurs	4	Erreur
0x6321	Paramètre manquant – Vérifier la fiche technique	4	Erreur
0x6322 à 0x634F	Réservé		
0x6350	Paramètre modifié – Vérifier la configuration	4	Erreur
0x6351 à 0x76FF	Réservé		
0x7700	Rupture d'un conducteur de dispositif subordonné – Vérifier l'installation	4	Erreur
0x7701 à 0x770F	Rupture d'un conducteur de dispositif subordonné 1 à 15 – Vérifier l'installation	4	Erreur
0x7710	Court-circuit – Vérifier l'installation	4	Erreur
0x7711	Défaut de mise à la terre – Vérifier l'installation	4	Erreur
0x7712 à 0x8BFF	Réservé		
0x8C00	Défaut de l'application spécifique à la technologie – Réinitialiser le Dispositif	4	Erreur
0x8C01	Simulation active – Vérifier le mode de fonctionnement	3	Avertissement
0x8C02 à 0x8C0F	Réservé		
0x8C10	Dépassement de plage de processus variable – Données de Processus peu sûres	2	Avertissement
0x8C11 à 0x8C1F	Réservé		
0x8C20	Dépassement de plage de mesure – Vérifier l'application	4	Erreur
0x8C21 à 0x8C2F	Réservé		
0x8C30	Plage de processus variable insuffisante – Données de Processus peu sûres	2	Avertissement
0x8C31 à 0x8C3F	Réservé		
0x8C40	Maintenance requise – Nettoyage	1	Notification
0x8C41	Maintenance requise – Remplissage	1	Notification
0x8C42	Maintenance requise – Remplacer les pièces d'usure	1	Notification
0x8C43 à 0x8C9F	Réservé		
0x8CA0 à 0x8DFF	Spécifique au fournisseur		
0x8E00 à 0xAFFF	Réservé		
0xB000 à 0xBFFF	Réservé aux profils		
0xC000 à 0xFEFF	Réservé		
0xFF00 à 0xFFFF	Codes d'Événements spécifiques à l'interface SDCI (voir le Tableau D.2)		

NOTE 1 Voir B.2.18.
NOTE 2 Voir le Tableau A.19.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau D.2 énumère les Événements SDCI de base liés à la Gestion Système, à l'application du Dispositif ou du Maître et spécifie la manière dont ils sont codés. D'autres types d'Événements peuvent être rapportés, mais ils ne sont pas spécifiés dans la présente norme. Le traitement de ces Événements par le Maître est spécifique au fournisseur.

Tableau D.2 – EventCodes SDCI de base

Incident [a]	Origine	Instance	Nom	EventCode	Action	Remarque
Gestion Système						
Indication du mode	LOCAL	DL	NEW_SLAVE	0xFF21	Arrêt des PD	Voir l'Article 11
Perte de communication du Dispositif	LOCAL	APP	DEV_COM_ LOST	0xFF22	-	Voir l'Article 11
Discordance d'identification du Stockage des Données	LOCAL	APP	DS_IDENT_ MISMATCH	0xFF23	-	Voir l'Article 11
Débordement de tampon de Stockage des Données	LOCAL	APP	DS_BUFFER _OVERFLOW	0xFF24	-	Voir l'Article 11
Accès aux paramètres de Stockage des Données refusé	LOCAL	APP	DS_ACCESS _DENIED	0xFF25	-	Voir l'Article 11
Non spécifié						
Signalement incorrect d'Événement	LOCAL	DL	EVENT	0xFF31	Event.ind	Voir l'Article 11
Spécifiques à l'application du Dispositif						
Demande de téléchargement amont de Stockage des Données	DISTANT	APP	DS_UPLOAD _REQ	0xFF91	Event.ind	
Réservé	DISTANT	APP		0xFF98	Event.ind	Ne doit pas être utilisé

[a] Tous les Événements ont un StatusCode de type 2 (avec détails), un EventQualifier de type "Notification", ou sont en mode EventQualifier "Occurrence unique".

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe E
(normative)

Types de Données

E.1 Généralités

La présente annexe décrit les types de données de base et composés. Les exemples démontrent les structures et les aspects liés à la transmission des types de données pour un usage singulier ou en paquets.

NOTE Des exemples supplémentaires sont donnés dans la référence [6].

E.2 Types de données de base

E.2.1 Généralités

Le codage de types de données de base est illustré uniquement pour un usage singulier, caractérisé par

- des Données de Processus constituées d'un type de données de base;
- des paramètres constitués d'un type de données de base;
- un accès au Sous-index (>0) pour des éléments de données individuels de paramètres de types de données composés (matrices, enregistrements).

E.2.2 BooleanT

Un type BooleanT représente un type de données qui peut prendre uniquement deux valeurs différentes, c'est-à-dire TRUE et FALSE. Le type de données est spécifié dans le Tableau E.1. Le codage pour un usage singulier est présenté dans le Tableau E.2. Un émetteur doit toujours utiliser 0xFF pour 'TRUE' ou 0x00 pour 'FALSE'. Un récepteur peut interpréter la plage allant de 0x01 à 0xFF comme étant 'TRUE' et doit interpréter 0x00 comme 'FALSE' pour simplifier les mises en œuvre. La forme en paquets est présentée dans le Tableau E.22 et la Figure E.8.

Tableau E.1 – BooleanT

Nom de type de données	Plage de valeurs	Résolution	Longueur
BooleanT	TRUE/FALSE	-	1 bit ou 1 octet

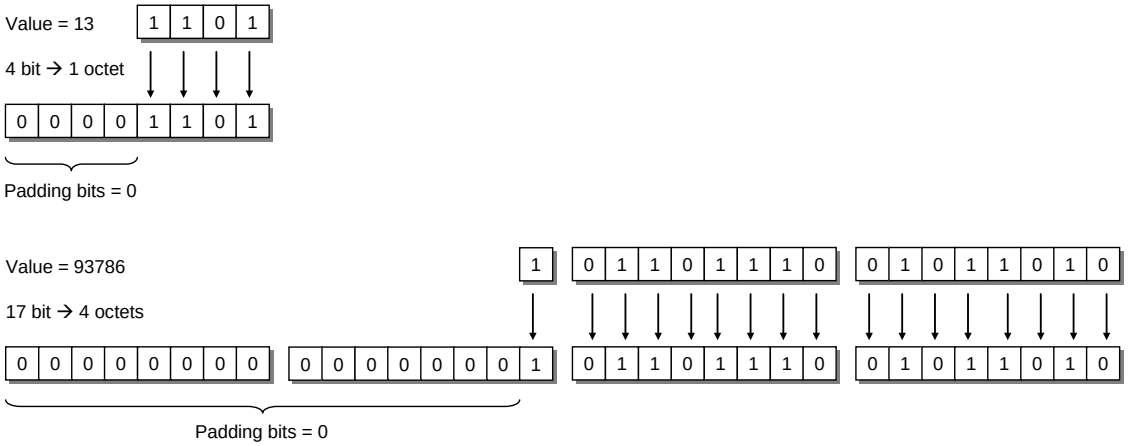
Tableau E.2 – Codage BooleanT

Bit	7	6	5	4	3	2	1	0	Valeurs
TRUE	1	1	1	1	1	1	1	1	0xFF
FALSE	0	0	0	0	0	0	0	0	0x00

E.2.3 UIntegerT

Un type UIntegerT représente un nombre non signé décrit par 2 à 64 bits ("énuméré"). Le nombre est placé et aligné à droite dans les conteneurs d'octets autorisés suivants: 1, 2, 4 ou 8. Les bits de remplissage de poids fort sont remplis par "0". Des exemples de codage sont illustrés en Figure E.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Value	Valeur
Padding bits	Bits de remplissage

Figure E.1 – Exemples de codage d'UIntegerT

Le type de données UIntegerT est spécifié dans le Tableau E.3 pour un usage singulier.

Tableau E.3 – UIntegerT

Nom de type de données	Plage de valeurs	Résolution	Longueur
UIntegerT	0 à $2^{\text{longueur binaire}} - 1$	1	1 octet, ou 2 octets, ou 4 octets, ou 8 octets
NOTE 1 Les bits de remplissage de poids fort sont remplis par "0". NOTE 2 L'octet de poids fort (MSO) est envoyé en premier.			

E.2.4 IntegerT

Un type IntegerT représente un nombre signé décrit par 2 à 64 bits. Le nombre est placé dans les conteneurs d'octets autorisés suivants: 1, 2, 4 ou 8, aligné et étendu à droite, correctement signé sur le nombre de bits choisis. Le type de données est spécifié dans le Tableau E.4 pour un usage singulier. SN représente le signe, "0" étant utilisé pour tous les nombres positifs et zéro, et "1" pour tous les nombres négatifs. Les bits de remplissage utilisent le contenu du bit de signe (SN).

Tableau E.4 – IntegerT

Nom de type de données	Plage de valeurs	Résolution	Longueur
IntegerT	$-2^{\text{longueur binaire } -1}$ à $2^{\text{longueur binaire } -1} - 1$	1	1 octet, ou 2 octets, ou 4 octets, ou 8 octets
NOTE 1 Les bits de remplissage de niveau élevé sont remplis avec la valeur du bit de signe (SN). NOTE 2 L'octet de poids fort (MSO) est envoyé en premier (numéro d'octet correspondant le plus petit dans le Tableau E.5).			

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Les 4 possibilités de codage dans des conteneurs sont énumérées du Tableau E.5 au Tableau E.8.

Tableau E.5 – Codage de IntegerT (8 octets)

Bit	7	6	5	4	3	2	1	0	Conteneur
Octet 1	SN	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	8 octets
Octet 2	2^{55}	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Octet 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Octet 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Tableau E.6 – Codage de IntegerT (4 octets)

Bit	7	6	5	4	3	2	1	0	Conteneur
Octet 1	SN	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	4 octets
Octet 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Tableau E.7 – Codage de IntegerT (2 octets)

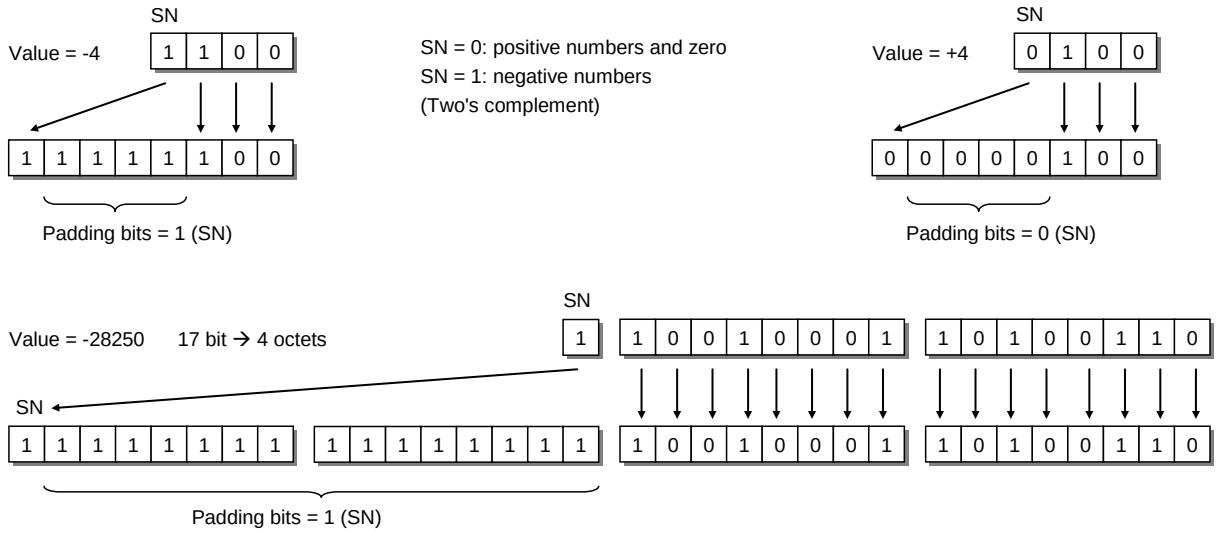
Bit	7	6	5	4	3	2	1	0	Conteneur
Octet 1	SN	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2 octets
Octet 2	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	

Tableau E.8 – Codage de IntegerT (1 octet)

Bit	7	6	5	4	3	2	1	0	Conteneur
Octet 1	SN	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	1 octet

Des exemples de codage dans des conteneurs sont illustrés en Figure E.2

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Value	Valeur
Padding bits	bits de remplissage
SN = 0: positive numbers and zero	SN = 0: nombres positifs et zéro
SN = 1: negative numbers	SN = 1: nombres négatifs
(Two's complement)	(Complément à deux)

Figure E.2 – Exemples de codage de IntegerT

E.2.5 Float32T

Un type Float32T représente un nombre spécifié par la norme IEEE 754-2008 comme étant à simple précision (32 bits). Le Tableau E.9 donne la définition et le Tableau E.10 fournit le codage de ce type. SN représente le signe, "0" étant utilisé pour tous les nombres positifs et zéro, et "1" pour tous les nombres négatifs.

Tableau E.9 – Float32T

Nom de type de données	Plage de valeurs	Résolution	Longueur
Float32T	Voir norme IEEE 754-2008	Voir norme IEEE 754-2008	4 octets

Tableau E.10 – Codage de Float32T

Bits	7	6	5	4	3	2	1	0
Octet 1	SN	Exposant (E)						
	2^0	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 2	(E)	Fraction (F)						
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Octet 3	Fraction (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Octet 4	Fraction (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Pour réaliser des valeurs d'exposant négatives, un mécanisme de codage spécial des exposants est mis en place de la manière suivante:

L'exposant (E) de Float32T est codé au moyen d'une représentation à décalage binaire, le décalage du zéro étant 127; également appelé "biais d'exposant" dans la norme IEEE 754-2008.

E_{min} = 0x01 - 0x7F = -126

E_{max} = 0xFE - 0x7F = 127

Biais d'exposant = 0x7F = 127

Ainsi, comme défini par la représentation à décalage binaire, pour obtenir un exposant vrai, le décalage de 127 doit être soustrait de l'exposant enregistré.

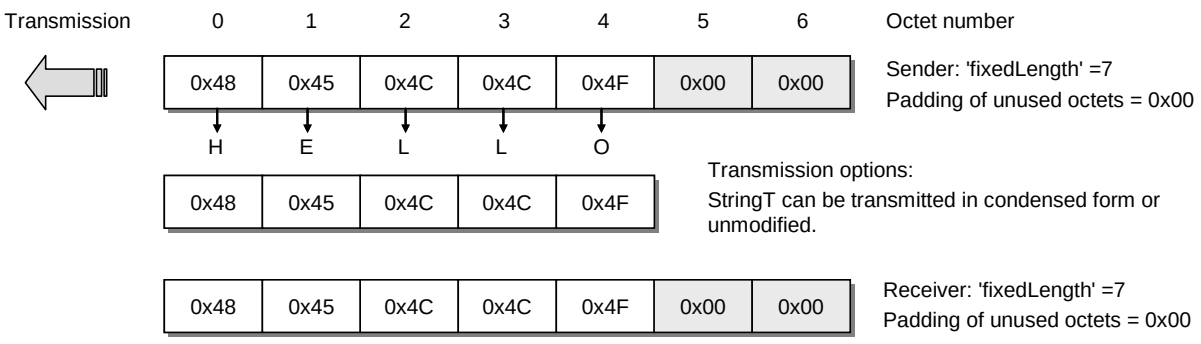
E.2.6 StringT

Un type StringT représente une séquence ordonnée de symboles (caractères) ayant une longueur d'octets variable ou fixe (au maximum 232 octets) codée en US-ASCII (7 bits) ou UTF-8. Le codage UTF-8 utilise un octet pour tous les caractères ASCII et jusqu'à 4 octets pour les autres caractères. 0x00 n'est pas autorisé en tant que caractère. La définition du type StringT est donnée dans le Tableau E.11.

Tableau E.11 – StringT

Nom de type de données	Codage	Normes	Longueur
StringT	US-ASCII	voir ISO/CEI 646	Toute longueur de chaîne de caractères d'au maximum 232 octets
	UTF-8	voir ISO/CEI 10646	
NOTE La longueur peut être obtenue à partir d'un IODD de Dispositif via l'attribut 'fixedLength'.			

Une instance de StringT peut être plus courte que celle définie par l'attribut "fixedLength" de l'IODD. 0x00 doit être utilisé pour le remplissage d'octets non utilisés. Les chaînes de caractères peuvent être transmises en taille réelle dans le cas d'un accès singulier (voir la Figure E.3). Pour l'optimisation de la transmission, il est possible d'omettre les octets de remplissage si l'attribut "fixedLengthRestriction" de l'IODD n'est pas établi. Le récepteur peut déduire la longueur initiale de la longueur de l'ISDU ou en recherchant le premier caractère NULL (0x00) (voir A.5.2 et A.5.3).



Légende

Anglais	Français
Transmission	Transmission
Octet number	Numéro d'octet

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Sender: 'fixedLength' =7	Émetteur: 'fixedLength' =7
Padding of unused octets = 0x00	Remplissage des octets non utilisés = 0x00
Transmission options:	Options de transmission:
StringT can be transmitted in condensed form or unmodified.	Le type StringT peut être transmis sous forme condensée ou non modifiée.
Receiver: 'fixedLength' =7	Récepteur 'fixedLength' =7
Padding of unused octets = 0x00	Remplissage des octets non utilisés = 0x00

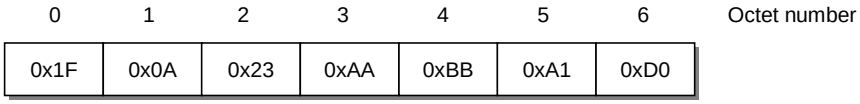
Figure E.3 – Accès singulier de StringT

E.2.7 OctetStringT

Un type OctetStringT représente une séquence ordonnée d'octets de longueur fixe (au maximum 232 octets). Le Tableau E.12 en donne la définition et la Figure E.4 fournit un exemple de codage pour une longueur fixe de 7.

Tableau E.12 – OctetStringT

Nom de type de données	Plage de valeurs	Normes	Longueur
OctetStringT	0x00 à 0xFF par octet	-	Longueur fixe d'au maximum 232 octets
NOTE La longueur peut être obtenue à partir d'un IODD de Dispositif via l'attribut ‘fixedLength'.			



Anglais	Français
Octet number	Numéro d'octet

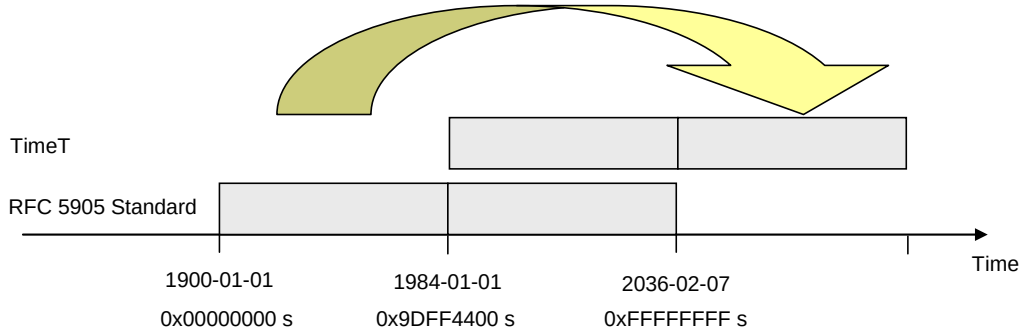
Figure E.4 – Exemple de codage de OctetStringT

E.2.8 TimeT

Un TimeT repose sur la norme RFC 5905 et est constitué de deux valeurs non signées qui expriment le temps réseau par rapport à une date particulière. Sa sémantique a été modifiée par rapport à la norme RFC 5905 conformément à la Figure E.5. Le Tableau E.13 donne la définition et le Tableau E.14 fournit le codage du type TimeT.

Le premier élément est un type de données "entier non signé de 32 bits" qui fournit le temps réseau en secondes depuis le 01-01-1900, à 0:h00,00 (UTC) ou depuis le 07-02-2036 à 6.28,16 (UTC) pour des valeurs de temps inférieures à 0x9DFF4400, qui représentent le 01-01-1984 à 0h00,00 (UTC). Le second élément est un type de données "entier non signé de 32 bits" qui fournit la partie fractionnaire des secondes en $1/2^{32}$ s. Les retournements au bout de 136 ans ne sont pas automatiquement détectables et doivent être maintenus par l'application.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
RFC 5905 Standard	Norme RFC 5905
Time	Temps

Figure E.5 – Définition de TimeT

Tableau E.13 – TimeT

Nom de type de données	Plage de valeurs	Résolution	Longueur
TimeT	Octet 1 à 4 (voir le Tableau E.14): $0 \le i \le (2^{32}-1)$	s (secondes)	8 Octets (entier non signé de 32 bits + entier non signé de 32 bits)
	Octet 5 à 8 (voir le Tableau E.14): $0 \le i \le (2^{32}-1)$	$(1/2^{32})$ s	
NOTE Les entiers non signés de 32 bits sont des types de données ordinaires en informatique.			

Tableau E.14 – Codage de TimeT

Bit	7	6	5	4	3	2	1	0	Définitions
Octet 1	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Secondes depuis le 01-01-1900, à 0.00,00 ou depuis le 07-02- 2036 à 6.28,16 lorsque la valeur du temps est inférieure à 0x9DFF4400.00000000
Octet 2	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 3	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 4	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	Partie fractionnaire des secondes. Une unité est égale à $1/(2^{32})$ s
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = bit de poids fort LSB = bit de poids faible

E.2.9 TimeSpanT

Un type TimeSpanT est une valeur entière de 64 bits, c'est-à-dire un nombre binaire en complément à deux d'une longueur de huit octets, fournissant la différence de temps réseau en parties fractionnaires de secondes ($1/2^{32}$ème de seconde). Le Tableau E.15 donne la définition et le Tableau E.16 fournit le codage du type TimeSpanT.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau E.15 – TimeSpanT

Nom de type de données	Plage de valeurs	Résolution	Longueur
TimeSpanT	Octet 1 à 8 (voir le Tableau E.16): - $2^{63} \leq i \leq (2^{63}-1)$	$(1/2^{32})$ s	8 octets (entier de 64 bits)
NOTE Les entiers non signés de 64 bits sont des types de données ordinaires en informatique.			

Tableau E.16 – Codage de TimeSpanT

Bit	7	6	5	4	3	2	1	0	Définitions
Octet 1	2^{63}	2^{62}	2^{61}	2^{60}	2^{59}	2^{58}	2^{57}	2^{56}	Partie fractionnaire des secondes comme entier de 64 bits. Une unité est égale à $1/(2^{32})$ s.
Octet 2	2^{55}	2^{54}	2^{53}	2^{52}	2^{51}	2^{50}	2^{49}	2^{48}	
Octet 3	2^{47}	2^{46}	2^{45}	2^{44}	2^{43}	2^{42}	2^{41}	2^{40}	
Octet 4	2^{39}	2^{38}	2^{37}	2^{36}	2^{35}	2^{34}	2^{33}	2^{32}	
Octet 5	2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	
Octet 6	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	
Octet 7	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	
Octet 8	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	
	MSB							LSB	MSB = bit de poids fort LSB = bit de poids faible

E.3 Types de données composés

E.3.1 Généralités

Les types de données composés sont des combinaisons de types de données de base uniquement. Un type de données composé est constitué de plusieurs types de données de base rassemblés dans une séquence d'octets. L'espace binaire non utilisé doit être rempli par "0".

E.3.2 ArrayT

Un Type ArrayT invoqué par un Index est une structure de données ayant des éléments de données du même type de données. Les éléments de données individuels sont adressables par le Sous-index. Le Sous-index 0 adresse l'ensemble de la matrice dans l'espace d'Index. Les règles de structuration des matrices sont données dans le Tableau E.17.

Tableau E.17 – Règles de structuration d'ArrayT

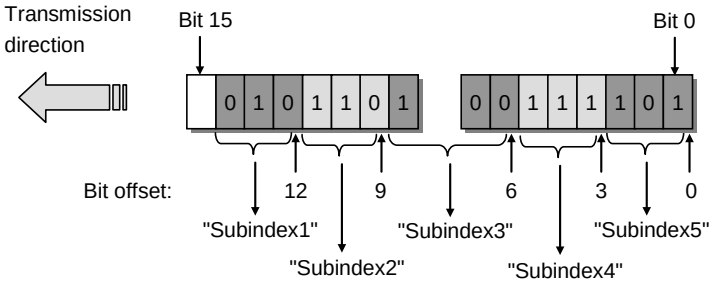
Numéro de règle	Spécification de règle
1	Les éléments de données de Sous-index sont assemblés dans une rangée sans espaces qui décrit une séquence d'octets
2	L'élément de données de Sous-index le plus élevé (n) est aligné à droite dans la séquence d'octets
3	UIntegerT et IntegerT d'une longueur ≥ 58 bits et < 64 bits ne sont pas autorisés

Le Tableau E.18 et la Figure E.6 donnent un exemple d'accès à une matrice. Son contenu est un jeu de paramètres du même type de données de base.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau E.18 – Exemple d'accès à un type ArrayT

Index	Sous-index	Décalage	Éléments de données	Type de Données
66	1	12	0x2	IntegerT, 'bitLength' = 3
	2	9	0x6	
	3	6	0x4	
	4	3	0x7	
	5	0	0x5	



Légende

Anglais	Français
Transmission direction	Sens de transmission
Bit offset:	Décalage binaire:
Subindex	Sous-index

Figure E.6 – Exemple d'une structure de données ArrayT

E.3.3 RecordT

Un enregistrement invoqué par un Index est une structure de données ayant des éléments de données de types différents. Le Sous-index permet l'adressage d'éléments de données individuels dans l'enregistrement sur certaines positions binaires.

NOTE Les positions binaires dans un type RecordT peuvent être obtenues à partir de l'IODD du Dispositif particulier.

Les règles de structuration des enregistrements sont données dans le Tableau E.19.

Tableau E.19 – Règles de structuration de RecordT

Numéro de règle	Spécification de règle
1	Les Sous-index dans l'IODD doivent être énumérés en ordre ascendant de 1 à *n* en décrivant une séquence d'octets. Les espaces dans la liste de Sous-index sont autorisés
2	Les décalages binaires doivent toujours être indiqués dans cette séquence d'octets (peut ne pas présenter d'ordre strict dans l'IODD)
3	Le décalage binaire commence par le dernier octet de la séquence; cet octet commence avec un décalage 0 pour le bit de poids faible et avec un décalage 7 pour le bit de poids fort
4	Les types de données suivants doivent toujours être alignés sur les frontières d'octets: Float32T, StringT, OctetStringT, TimeT et TimeSpanT
5	Les UIntegerT et IntegerT d'une longueur ≥ 58 bits doivent toujours être alignés d'un côté d'une frontière d'octet
6	Il est fortement recommandé que les types UIntegerT et IntegerT d'une longueur ≥ 8 bits soient toujours alignés d'un côté d'une frontière d'octet

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Numéro de règle	Spécification de règle
7	Il est fortement recommandé que les types UIntegerT et IntegerT d'une longueur < 8 bits ne traversent pas les frontières d'octets
8	Une position binaire donnée ne doit pas être utilisée par plus d'un élément d'enregistrement

Le Tableau E.20 donne un exemple 1 pour l'accès à un RecordT. Il est constitué de divers paramètres appelés "État", "Texte" et "Valeur".

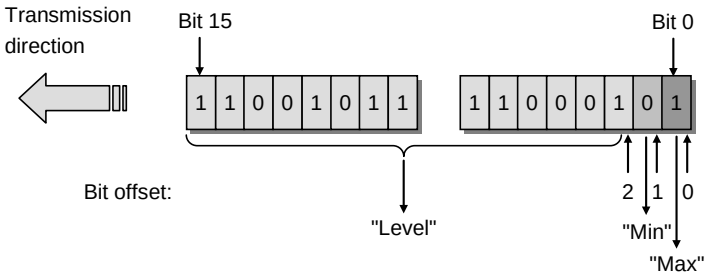
Tableau E.20 – Premier exemple (1) d'accès à un RecordT

Index	Sous-index	Décalage	Éléments de données							Type de Données	Nom
47	1	88	0x23	0x45						UIntegerT, 'bitLength' = 16	État
	2	32	H	E	L	L	O	0x00	0x00	StringT, 'fixedLength' = 7	Texte
	3	0	0x56	0x12	0x22	0x34				UIntegerT, 'bitLength = 32	Valeur
NOTE Les valeurs 'bitLength' et 'fixedLength' sont définies dans l'IODD du Dispositif particulier.											

Le Tableau E.21 donne un exemple (2) pour l'accès à un RecordT. Il est constitué de divers paramètres appelés "Niveau", "Min" et "Max". La Figure E.7 illustre la structure de données correspondante.

Tableau E.21 – Deuxième exemple (2) d'accès à un RecordT

Index	Sous-index	Décalage	Éléments de données			Type de Données	Nom
46	1	2	0x32	0xF1		UIntegerT, 'biLlength' = 14	Niveau
	2	1	FALSE			BooleanT	Min
	3	0	TRUE			BooleanT	Max.
NOTE La 'bitLength' est définie dans l'IODD du Dispositif particulier.							



Légende

Anglais	Français
Transmission direction	Sens de transmission
Bit offset:	Décalage binaire:
"Level"	"Niveau"

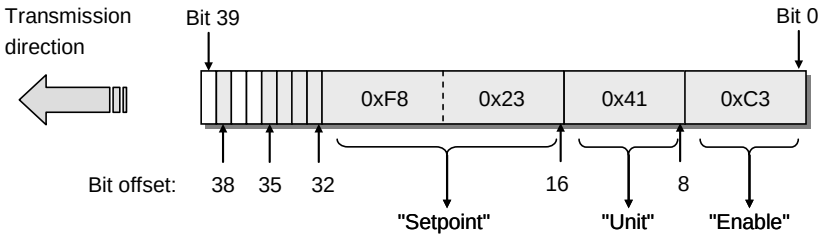
Figure E.7 – Deuxième exemple (2) d'une structure de RecordT

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Le Tableau E.22 donne un exemple (3) pour l'accès à un RecordT. Il est constitué de divers paramètres appelés "Contrôle" jusqu'à "Activation". La Figure E.8 représente la structure RecordT correspondant au troisième exemple (3) avec les décalages binaires.

Tableau E.22 – Troisième exemple (3) d'accès à un RecordT

Index	Sous-index	Décalage	Éléments de données			Type de Données	Nom
45	1	32	TRUE			BooleanT	NewBit
	2	33	FALSE			BooleanT	DR4
	3	34	FALSE			BooleanT	CR3
	4	35	TRUE			BooleanT	CR2
	5	38	TRUE			BooleanT	Contrôle
	6	16	0xF8	0x23		OctetStringT, 'fixedLength' = 2	point de consigne
	7	8	0x41			StringT, 'fixedLength' = 1	Unité
	8	0	0xC3			OctetStringT, 'fixedLength' = 1	Activation
NOTE La 'fixedLength' est définie dans l'IODD du Dispositif particulier.							



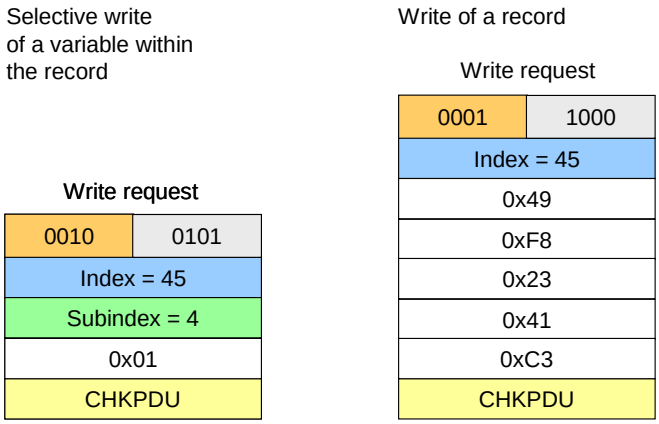
Légende

Anglais	Français
Transmission direction	Sens de transmission
Bit offset:	Décalage binaire:
"Setpoint"	“Point de consigne”
"Unit"	"Unité"
"Enable"	"Activation"

Figure E.8 – Troisième exemple (3) d'une structure de RecordT

La Figure E.9 illustre une demande d'écriture sélective d'une variable dans le RecordT du troisième exemple (3) et une demande d'écriture de l'ensemble RecordT complet (voir A.5.7).

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



Légende

Anglais	Français
Selective write of a variable within the record	Écriture sélective d'une variable dans l'enregistrement
Write of a record	Écriture d'un enregistrement
Write request	Demande d'écriture
Subindex	Sous-index

Figure E.9 – Demandes d'écriture pour l'exemple 3

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe F
(normative)

Structure de l'objet de données "Stockage de Données"

Le Tableau F.1 décrit la structure d'un objet de données Stockage des Données(DS) dans le Maître (voir 11.3.2).

Tableau F.1 – Structure de l'objet de données DS enregistré

Partie	Nom du paramètre	Définition	Type de Données
Objet 1	ISDU_Index	Index des ISDU (0 à 65 535)	Unsigned16
	ISDU_Subindex	Index des ISDU (0 à 255)	Unsigned8
	ISDU_Length	Longueur de l'enregistrement ultérieur	Unsigned8
	ISDU_Data	Enregistrement de longueur ISDU_Length	Enregistrement
Objet 2	ISDU_Index	Index des ISDU (0 à 65 535)	Unsigned16
	ISDU_Subindex	Index des ISDU (0 à 255)	Unsigned8
	ISDU_Length	Longueur de l'enregistrement ultérieur	Unsigned8
	ISDU_Data	Enregistrement de longueur ISDU_Length	Enregistrement
----------			
Objet *n*	ISDU_Index	Index des ISDU (0 à 65 535)	Unsigned16
	ISDU_Subindex	Index des ISDU (0 à 255)	Unsigned8
	ISDU_Length	Longueur de l'enregistrement ultérieur	Unsigned8
	ISDU_Data	Enregistrement de longueur ISDU_Length	Enregistrement

Le Dispositif doit calculer la taille de mémoire requise en récapitulant les objets 1 à *n* (voir le Tableau B.10, Sous-index 3).

Le Maître doit enregistrer localement, dans une mémoire non volatile, les informations d'en-tête spécifiées dans le Tableau F.2. Voir le Tableau B.10.

Tableau F.2 – Informations d'en-tête associées aux objets de données DS enregistrés

Partie	Nom du paramètre	Définition	Type de Données
En-tête	Parameter Checksum	Signature CRC 32 bits ou compteur de versions (voir 10.4.8)	Unsigned32
	VendorID	Voir B.1.8	Unsigned16
	DeviceID	Voir B.1.9	Unsigned32
	FunctionID	Voir B.1.10	Unsigned16

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe G
(normative)

Conformité du Maître et du Dispositif

G.1 Exigences de compatibilité électromagnétique (CEM)

G.1.1 Généralités

Les exigences de compatibilité électromagnétique de la présente annexe sont uniquement pertinentes pour la partie interface SDCI d'un Maître ou d'un Dispositif particulier. Les fonctions technologiques d'un Dispositif ainsi que ses exigences CEM pertinentes ne relèvent pas du domaine d'application de la présente norme. À cet égard, les normes de produit spécifiques au Dispositif doivent s'appliquer. En général, pour le Maître, les exigences CEM pour les périphériques sont spécifiées dans la CEI 61131-2 ou dans la CEI 61000-6-2.

Pour garantir des conditions de fonctionnement correctes de l'interface SDCI, les configurations d'essai spécifiées au G.1.6 (Maître) ou G.1.7 (Dispositif) doivent être maintenues pendant toute la durée des essais CEM. Les essais exigés dans la norme de produit de l'appareil en essai (EUT – equipment under test) peuvent également être effectués en mode SIO.

G.1.2 Conditions de fonctionnement

Il est fortement recommandé d'évaluer l'interface SDCI pendant la phase de démarrage en utilisant les durées de cycle données dans le Tableau G.1. Dans la plupart des cas, ceci implique les exigences de durée minimale d'exécution de ces essais. L'interface SDCI peut également être évaluée pendant le fonctionnement normal du Dispositif, sous réserve de la réalisation du nombre requis de séquences M, spécifié dans le Tableau G.1, lors de chaque essai.

G.1.3 Critères de performances

a) Critère de performance A

L'interface SDCI, fonctionnant à une durée de cycle moyenne, comme spécifié dans le Tableau G.1, ne doit pas présenter plus de six erreurs de séquences M détectées au sein du nombre de séquences M donné dans le Tableau G.1. Aucune interruption de la communication n'est admise.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Tableau G.1 – Conditions d'essai CEM pour l'interface SDCI

Vitesse de transmission	Maître		Dispositif		Nombre maximal d'erreurs de séquence M
	t_{CYC}	Nombre de séquences M de TYPE_2_5 (lecture) (6 octets)	t_{CYC}	Nombre de séquences M de TYPE_0 (lecture) (4 octets)	
4,8 kbit/s	18,0 ms	300 (6 000)	100 T_{BIT} (20,84 ms)	350 (7 000)	6
38,4 kbit/s	2,3 ms	450 (9 000)	100 T_{BIT} (2,61 ms)	500 (10 000)	6
230,4 kbit/s	0,4 ms	700 (14 000)	100 T_{BIT} (0,44 ms)	800 (16 000)	6
NOTE Le nombre de séquences M est calculé conformément à l'algorithme donné en H.2 et arrondi. Le plus grand nombre de séquences M (entre parenthèses) est exigé si un certain essai (par exemple, des transitoires rapides en salves) applique des perturbations uniquement à un certain rapport salve/cycle (voir le Tableau G.2).					

b) Critère de performance B

Le taux d'erreur du critère A doit également être satisfait après, mais non pendant l'essai. Aucune modification de l'état de fonctionnement courant (par exemple perte permanente de communication) ou des données enregistrées n'est admise.

G.1.4 Essais d'immunité exigés

Le Tableau G.2 définit les essais CEM à effectuer.

Tableau G.2 – Niveaux d'essais CEM

Phénomènes	Niveau d'essai	Critère de performance	Contraintes
Décharges électrostatiques (DES) CEI 61000-4-2	Décharge dans l'air: ± 8 kV Décharge de contact: ± 4 kV	B	Voir G.1.4, a)
Champ électromagnétique aux fréquences radioélectriques. Modulé en amplitude CEI 61000-4-3	80 MHz - 1 000 MHz 10 V/m 1 400 MHz - 2 000 MHz 3 V/m 2 000 MHz - 2 700 MHz 1 V/m	A	Voir G.1.4, a) et G.1.4, b)
Transitoires rapides (en salves) CEI 61000-4-4	± 1 kV	A	5 kHz uniquement. Du fait du rapport salve/cycle de 15 ms/300 ms, le nombre de séquences M du Tableau G.1 doit être augmenté d'un facteur de 20. Voir G.1.4, c)
	± 2 kV	B	
Ondes de choc CEI 61000-4-5	Non exigé pour une liaison SDCI (la liaison SDCI est limitée à 20 m)		-
Fréquence radioélectrique en mode commun CEI 61000-4-6	0,15 MHz - 80 MHz 10 V f.é.m.	A	Voir G.1.4, b) et G.1.4, d)
Creux de tension et interruptions de tension CEI 61000-4-11	Non exigé pour une liaison SDCI		

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Les exigences suivantes s'appliquent également comme cela est spécifié dans le Tableau G.2.

a) Sachant que ce phénomène affecte l'ensemble du dispositif en essai, une norme existante de produit spécifique au Dispositif doit prévaloir sur les niveaux d'essai spécifiés ici.

b) L'essai doit être effectué par incréments de 1 % et un temps de maintien de 1 s. S'il est détecté une seule erreur de séquence M, à une certaine fréquence, cette fréquence doit être soumise à l'essai jusqu'à ce que le nombre de séquences M indiqué dans le Tableau G.1 ait été transmis ou jusqu'à ce que 6 erreurs de séquences M aient eu lieu.

c) La durée d'essai varie en fonction de la vitesse de transmission. La durée d'essai doit être d'au moins une minute (en augmentant en conséquence le nombre de séquences M transmises et le nombre d'erreurs autorisées).

d) Il est prévu que ce phénomène affecte très probablement le traitement interne des signaux analogiques des EUT, ainsi que, avec une très faible probabilité, la fonctionnalité de la communication SDCI. Par conséquent, une norme existante de produit spécifique au Dispositif doit prévaloir sur les niveaux d'essai spécifiés ici.

G.1.5 Essais d'émission exigés

La définition des limites d'émission ne fait pas l'objet de la présente norme. Les exigences de normes de familles de produits spécifiques au Dispositif ou de normes génériques s'appliquent; en général pour les environnements industriels, il s'agit de la CEI 61000-6-4.

Tous les essais d'émission doivent être effectués à la vitesse de communication la plus rapide possible, avec la durée de cycle la plus rapide.

G.1.6 Configurations d'essai de Maître

G.1.6.1 Règles générales

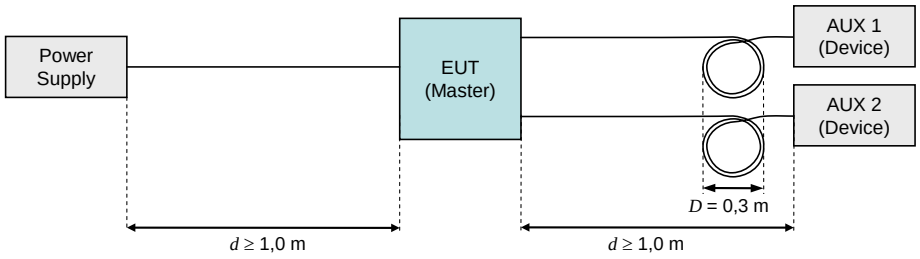
Les règles suivantes s'appliquent aux essais des Maîtres:

- Dans les schémas de montage d'essai ci-dessous, seuls l'interface SDCI et les câbles d'alimentation sont illustrés. Tous les autres câbles doivent être traités selon les exigences de la norme de produit pertinente.
- La mise à la terre du Maître et des Dispositifs doit être réalisée conformément à la norme ou au manuel de produit pertinent.
- Sauf indication contraire, le câble SDCI doit avoir une longueur totale de 20 m. Le cas échéant, une longueur supplémentaire posée comme une boucle d'induction, d'un diamètre de 0,3 m, peut être montée à 0,1 m au-dessus de la terre de référence.
- Le cas échéant, les Dispositifs auxiliaires doivent être placés 10 cm au dessus de la terre de référence (RefGND).
- Une configuration d'essai type comprend le Maître et deux Dispositifs, sauf pour l'essai de RF en mode commun pour lequel on doit utiliser un seul Dispositif.
- Chaque port doit satisfaire aux exigences CEM.

G.1.6.2 Décharges électrostatiques

La Figure G.1 illustre le montage d'essai de décharges électrostatiques conformément à la CEI 61000-4-2.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



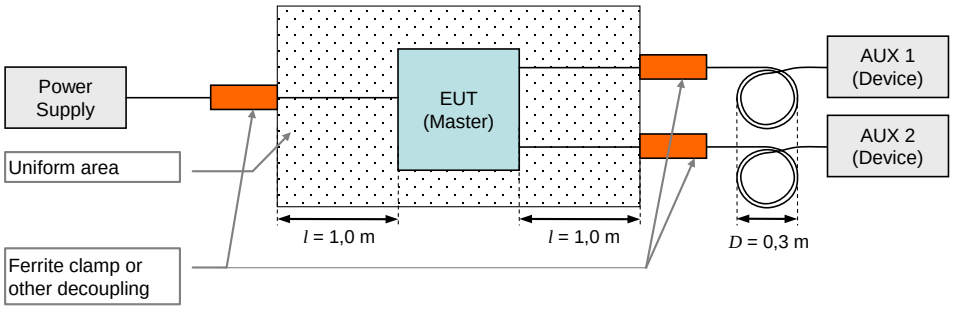
Légende

Anglais	Français
Power Supply	Alimentation
EUT (Master)	EUT (Maître)
(Device)	(Dispositif)

Figure G.1 – Montage d'essai de décharges électrostatiques (Maître)

G.1.6.3 Champ électromagnétique aux fréquences radioélectriques

La Figure G.2 illustre le montage d'essai de champ électromagnétique aux fréquences radioélectriques conformément à la CEI 61000-4-3.



Légende

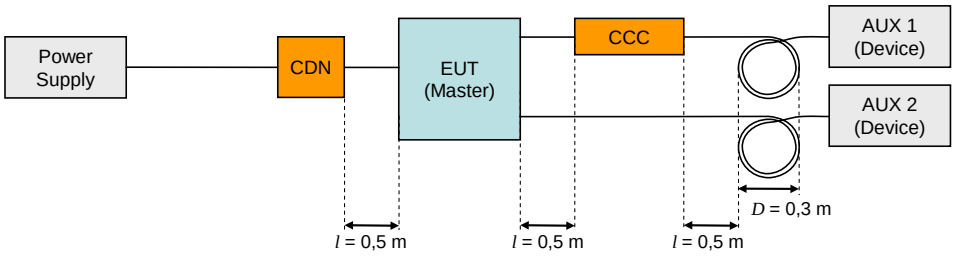
Anglais	Français
Power Supply	Alimentation
Uniform area	Zone uniforme
Ferrite clamp or other decoupling	Pince de ferrite ou autre élément de découplage
EUT (Master)	EUT (Maître)
(Device)	(Dispositif)

Figure G.2 – Montage d'essai de champ électromagnétique aux fréquences radioélectriques (Maître)

G.1.6.4 Transitoires rapides (en salves)

La Figure G.3 illustre le montage d'essai de transitoires rapides, conformément à la CEI 61000-4-4. Aucun couplage entre la ligne SDC et AUX 2 n'est exigé.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



CDN: Réseau de couplage/découplage
CCC: Pince de couplage capacitif

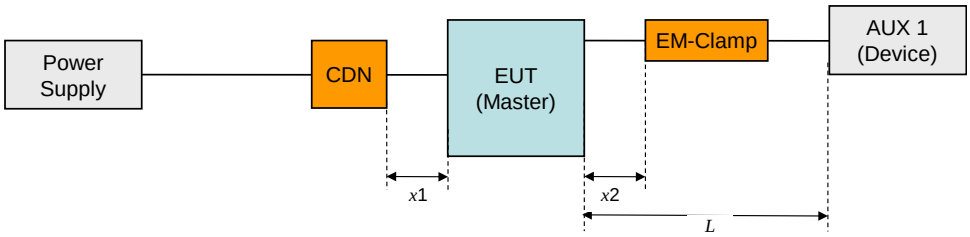
Légende

Anglais	Français
Power Supply	Alimentation
EUT (Master)	EUT (Maître)
(Device)	(Dispositif)

Figure G.3 – Montage d'essai de transitoires rapides (Maître)

G.1.6.5 Fréquences radioélectriques en mode commun

La Figure G.4 illustre le montage d'essai de fréquences radioélectriques en mode commun, conformément à la CEI 61000-4-6.



0,1 m ≤ ×1 ≤ 0,3 m
0,1 m ≤ × 2 ≤ 0,3 m
L = 1,0 m ± 0,05 m

Légende

Anglais	Français
Power Supply	Alimentation
EUT (Master)	EUT (Maître)
EM-Clamp	Pince EM
(Device)	(Dispositif)

Figure G.4 – Montage d'essai de fréquences radioélectriques en mode commun (Maître)

G.1.7 Configurations d'essai de Dispositifs

G.1.7.1 Règles générales

Les règles suivantes s'appliquent aux essais des Dispositifs:

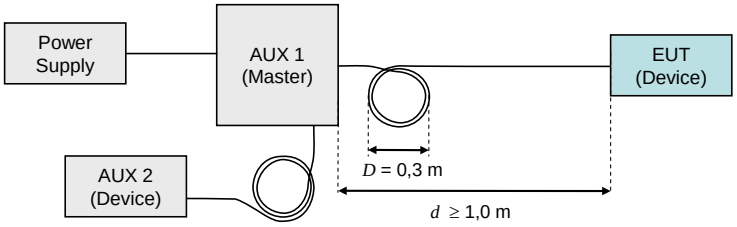
- Dans les schémas de montage d'essai ci-dessous, seuls l'interface SDCI et les câbles d'alimentation sont illustrés. Tous les autres câbles doivent être traités selon les exigences de la norme de produit pertinente.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

- La mise à la terre du Maître et des Dispositifs doit être réalisée conformément à la norme de produit pertinente ou au manuel d'utilisateur correspondant.
- Sauf indication contraire, le câble SDCI doit avoir une longueur totale de 20 m. Le cas échéant, une longueur supplémentaire posée comme une boucle d'induction, d'un diamètre de 0,3 m, peut être montée à 0,1 m au-dessus de la terre de référence (RefGND).
- Le cas échéant, les Dispositifs auxiliaires doivent être placés 10 cm au dessus de la terre de référence (RefGND).
- L'essai avec Dispositif en AUX 2 est facultatif.

G.1.7.2 Décharges électrostatiques

La Figure G.5 illustre le montage d'essai de décharges électrostatiques conformément à la CEI 61000-4-2.



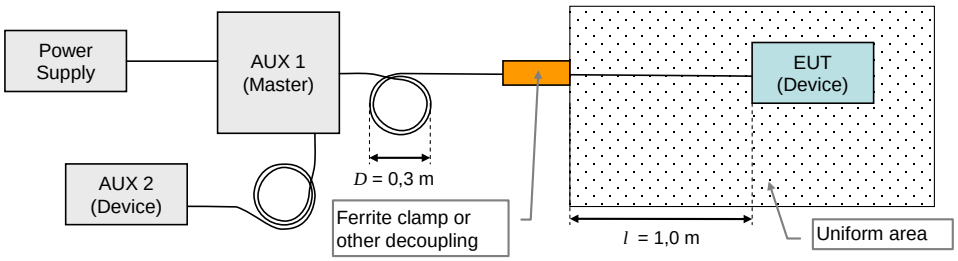
Légende

Anglais	Français
Power Supply	Alimentation
(Master)	(Maître)
EUT (Device)	EUT (Dispositif)
AUX 2 (Device)	AUX 2 (Dispositif)

Figure G.5 – Montage d'essai de décharges électrostatiques (Dispositif)

G.1.7.3 Champ électromagnétique aux fréquences radioélectriques

La Figure G.6 illustre le montage d'essai de champ électromagnétique aux fréquences radioélectriques conformément à la CEI 61000-4-3.



Légende

Anglais	Français
Power Supply	Alimentation
(Master)	(Maître)
EUT (Device)	EUT (Dispositif)
AUX 2 (Device)	AUX 2 (Dispositif)

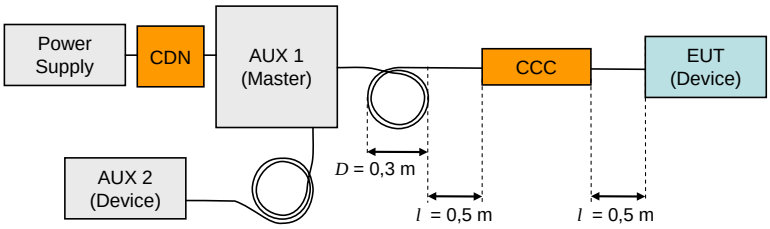
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Anglais	Français
Ferrite clamp or other decoupling	Pince de ferrite ou autre élément de découplage
Uniform area	Zone uniforme

Figure G.6 – Montage d'essai de champ électromagnétique aux fréquences radioélectriques (Dispositif)

G.1.7.4 Transitoires rapides (en salves)

La Figure G.7 illustre le montage d'essai de transitoires rapides, conformément à la CEI 61000-4-4.



CDN: Réseau de couplage/découplage, ici uniquement utilisé pour le découplage;
CCC: Pince de couplage capacitif

Légende

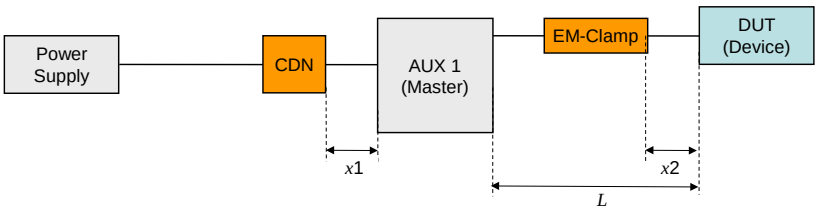
Anglais	Français
Power Supply	Alimentation
(Master)	(Maître)
EUT (Device)	EUT (Dispositif)
(Device)	(Dispositif)

Figure G.7 – Montage d'essai de transitoires rapides (Dispositif)

G.1.7.5 Fréquences radioélectriques en mode commun

La Figure G.8 illustre le montage d'essai de fréquences radioélectriques en mode commun, conformément à la CEI 61000-4-6.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe



0,1 m ≤ × 1 ≤ 0,3 m
0,1 m ≤ × 2 ≤ 0,3 m
L = 1,0 m ± 0,05 m

Légende

Anglais	Français
Power Supply	Alimentation
(Master)	(Maître)
DUT (Device)	DUT (Dispositif)
EM-clamp	pince EM

Figure G.8 – Montage d'essai de fréquences radioélectriques en mode commun (Dispositif)

G.2 Stratégies d'essais de conformité

G.2.1 Essai d'un Dispositif

Le Maître en AUX 1 (voir la Figure G.5) doit envoyer en continu un message de séquence M de TYPE _0 (Lecture de page 2 de Paramètres Directs) à la durée de cycle spécifiée dans le Tableau G.1 et compter les réponses manquantes et erronées du Dispositif. Les deux valeurs obtenues doivent être additionnées et indiquées.

NOTE Des instructions détaillées d'essais du Dispositif sont spécifiées dans la référence [9].

G.2.2 Essai d'un Maître

Le Dispositif en AUX 1 (voir la Figure G.1) doit utiliser la séquence M de Type_2_5. Ses Données de Processus d'entrée doivent être générées par un générateur aléatoire ou pseudo-aléatoire de 8 bits. Le Maître doit copier les Données de Processus d'entrée de tout message de Dispositif reçu vers les Données de Processus de sortie de message de Maître suivant à envoyer. La durée de cycle doit être conforme au Tableau G.1. Le Dispositif en AUX 1 doit comparer les Données de Processus de sortie aux Données de Processus d'entrée précédemment envoyées et compter le nombre de divergences. Le Dispositif doit également compter le nombre de messages perturbés manquants (non reçus pendant la durée de cycle prévue) ou reçus du Maître. Toutes les valeurs obtenues doivent être additionnées et indiquées.

NOTE 1 Une divergence entre Données de Processus envoyées et reçues indique à AUX1 que l'EUT (Maître) n'a pas reçu le message de Dispositif.

NOTE 2 Des instructions détaillées d'essais du Maître sont spécifiées dans la référence [9].

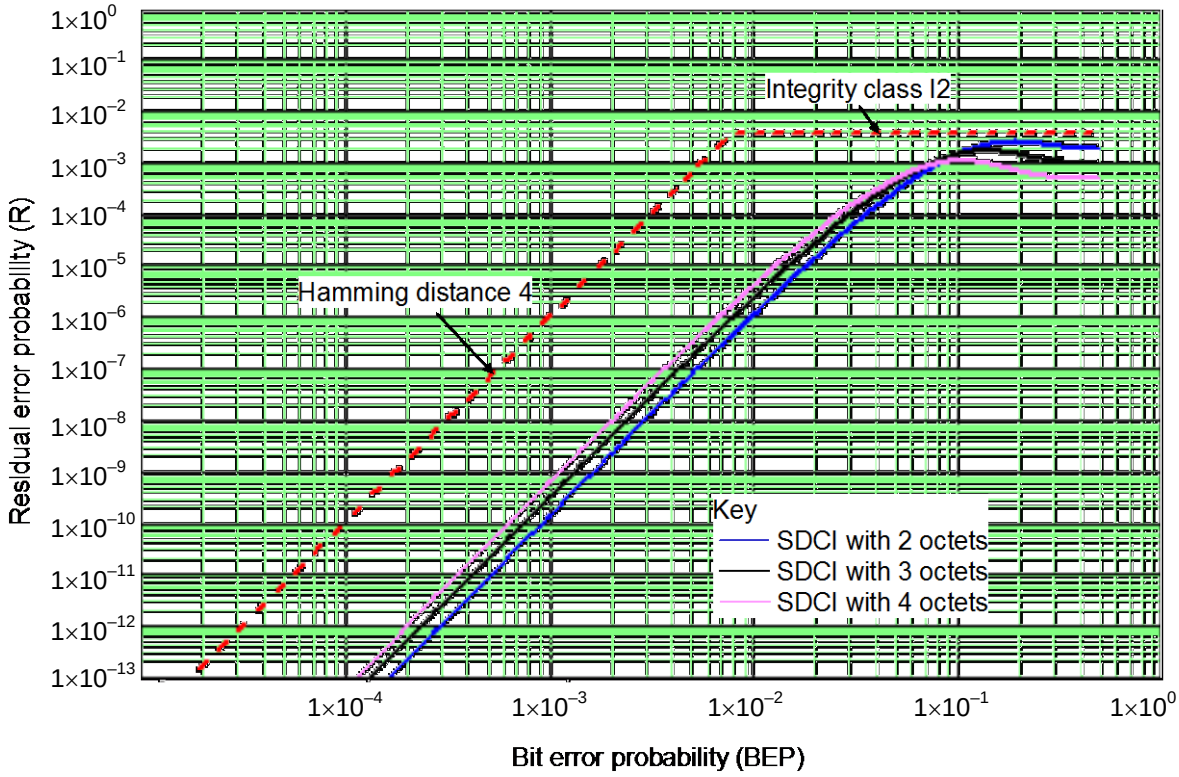
Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe H
(informative)

Probabilités d'erreurs résiduelles

H.1 Probabilité d'erreurs résiduelles du mécanisme d'intégrité des données SDCI

La Figure H.1 illustre la probabilité d'erreurs résiduelles (REP) du mécanisme d'intégrité des données SDCI qui est constitué de la procédure de vérification d'intégrité des données par la somme de contrôle ("XOR6"), comme spécifié en A.1.6, et de la parité UART. Le schéma fait référence à la CEI 60870-5-1 et à sa Classe I2 d'intégrité des données pour une distance de Hamming minimale de 4 (ligne en pointillés rouge).



Légende

Anglais	Français
Residual error probability (R)	Probabilité d'erreurs résiduelles (R)
Hamming distance 4	Distance de Hamming 4
Integrity class I2	Classe d'intégrité I2
Key	Légende
SDCI with 2 octets	SDCI avec 2 octets
SDCI with 3 octets	SDCI avec 3 octets
SDCI with 4 octets	SDCI avec 4 octets
Bit error probability (BEP)	Probabilité d'erreur sur les bits (BEP)

Figure H.1 – Probabilité d'erreurs résiduelles du mécanisme d'intégrité des données SDCI

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

La ligne de couleur bleue représente la courbe d'erreurs résiduelles d'une longueur de données de 2 octets. La courbe de couleur noire représente les erreurs résiduelles d'une longueur de données de 3 octets. La courbe de couleur violette représente les erreurs résiduelles d'une longueur de données de 4 octets.

H.2 Conditions d'essais CEM dérivées

Le critère de performance A, en G.1.3 est induit des exigences spécifiées dans la CEI 61158-2 en termes de susceptibilité au brouillage et de taux d'erreurs (citation; dans la présente norme, le terme "*trames*" devient "messages"):

- *Uniquement 1 trame erronée non détectée en 20 ans, à 1 600 trames/s;*
- *Le rapport entre trames non détectées et trames détectées ne doit pas dépasser 10^{-6};*
- *Les essais CEM ne doivent pas déceler plus de 6 trames erronées dans 100 000 trames.*

Pour l'interface SDCI, la première exigence se traduit par l'Equation (H.1). Cette équation permet de déterminer une valeur de la probabilité d'erreurs sur les bits (BEP). L'équation peut être résolue numériquement.

$$F20 \times R(BEP) \leq 1 \qquad \text{(H.1)}$$

où

F20 est le nombre de messages en 20 ans;

R(BEP) est la probabilité d'erreurs résiduelles du mécanisme de somme de contrôle et de parité (Figure H.1);

BEP est la probabilité d'erreurs sur les bits de la Figure H.1.

L'objectif de l'essai CEM est de démontrer que la BEP de la communication SDCI est conforme à la valeur déterminée au cours de la première étape. Pour des raisons pratiques, le nombre maximal de messages perturbés détectés est de 6. Le nombre de messages d'essai SDCI exigé peut être déterminé au moyen de l'Equation (H.2) et de la valeur de BEP déterminée au cours de la première étape.

$$NoTF \geq \frac{1}{BEP} \times \frac{1}{BitPerF} \times NopErr \qquad \text{(H.2)}$$

où

NoTF est le nombre de messages d'essai;

BitPerF est le nombre de bits par message;

NopErr est le nombre maximal de messages perturbés détectés = 6.

L'équation (H.2) est uniquement valide si l'on suppose que les messages ayant 1 erreur binaire sont plus fréquents que les messages ayant plusieurs erreurs. Une séquence M est constituée de deux messages. Par conséquent, le nombre calculé de messages d'essai est à diviser par 2 pour obtenir le nombre de séquences M du Tableau G.1.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe I
(informative)

Exemple d'une séquence de transmission d'ISDU

La Figure I.1 donne un exemple de transmission d'ISDU en utilisant un service AL_Read avec un Index de 16 bits et un Sous-index pour 19 octets de données utilisateur mises en correspondance avec une séquence M de TYPE_2_5 pour capteurs et avec interruption en cas de transmission d'Événements.

Master								Device			
		FC		CKT		PD	OD	OD	PD	CKS	
comment (state, action) (see in Table 46)	cycle nr	R W	Com Chan.	Flow CTRL	Frame Typ	CHK	Process Data	OnReq Data Master	OnReq Data Device	Process Data	CHK E PD	comment (state, action)
		1bit	2bit	5bit	2bit	6bit	8bit	8bit	8bit			
Idle_1	0	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	OnReq idle
ISDURequest_2, transmission,	1	0111	0000		10	xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	2	0110	0001		10	xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	3	0110	0010		10	xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	4	0110	0011		10	xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	5	0110	0100		10	xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	6	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	7	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	8	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	9	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	10	1111	0000		10	xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	11	1111	0000		10	xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	12	1110	0001		10	xxxxxx	xxxxxxxx		0001 0011	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	13	1110	0010		10	xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	14	1110	0011		10	xxxxxx	xxxxxxxx		Data 2	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	15	1110	0100		10	xxxxxx	xxxxxxxx		Data 3	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	16	1110	0101		10	xxxxxx	xxxxxxxx		Data 4	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	17	1110	0110		10	xxxxxx	xxxxxxxx		Data 5	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	18	1110	0111		10	xxxxxx	xxxxxxxx		Data 6	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	19	1110	1000		10	xxxxxx	xxxxxxxx		Data 7	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, no response, retry in next cycle	20	1110	1001		10	Err	xxxxxxxx				xxxxxx	ISDUResponse_4, korrupted CHK, don' t send resp.
ISDUResponse_4, no response, retry in next cycle	21	1110	1001		10	Err	xxxxxxxx				xxxxxx	ISDUResponse_4, corrupted CHK, don' t send resp.
ISDUResponse_4, reception	22	1110	1001		10	xxxxxx	xxxxxxxx		Data 8	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	34	1110	1010		10	xxxxxx	xxxxxxxx		Data 9	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception, start eventhandler	35	1110	1011		10	xxxxxx	xxxxxxxx		Data 10	xxxxxxxx	1 0 xxxxxx	ISDUResponse_4, transmission, freeze event
Read_Event_2, reception	36	1100	0000		10	xxxxxx	xxxxxxxx		Diag State with detail	xxxxxxxx	1 0 xxxxxx	Read_Event_2, transmission
Read_Event_2, reception	37	110x	xxxx		10	xxxxxx	xxxxxxxx		Event qualifier	xxxxxxxx	1 0 xxxxxx	Read_Event_2, transmission
Command handler_2, transmission set PDOutdata state to invalid	38	0010	0000		10	xxxxxx	xxxxxxxx	1001 1001		xxxxxxxx	1 0 xxxxxx	ComandHandler_2, reception, set PDOutdata state to invalid
Read_Event_2, reception	39	110x	xxxx		10	xxxxxx	xxxxxxxx		ErrorCode msb	xxxxxxxx	1 0 xxxxxx	Read_Event_2, transmission
Read_Event_2, reception	40	110x	xxxx		10	xxxxxx	xxxxxxxx		ErrorCode lsb	xxxxxxxx	1 0 xxxxxx	Read_Event_2, transmission
EventConfirmation_4, transmission, event handler idle	41	0100	0000		10	xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	0 0 xxxxxx	EventConfirmation, reception
ISDUResponse_4, reception	42	1110	1100		10	xxxxxx	xxxxxxxx		Data 11	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	43	1110	1101		10	xxxxxx	xxxxxxxx		Data 12	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	44	1110	1110		10	xxxxxx	xxxxxxxx		Data 13	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	45	1110	1111		10	xxxxxx	xxxxxxxx		Data 14	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	46	1110	0000		10	xxxxxx	xxxxxxxx		Data 15	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	47	1110	0001		10	xxxxxx	xxxxxxxx		Data 16	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	48	1110	0010		10	xxxxxx	xxxxxxxx		CHKPDU	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
Idle_1	49	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	50	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	51	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	52	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Write Parameter, transmission	53	0011	0000		10	xxxxxx	xxxxxxxx	xxxxxxxx		xxxxxxxx	0 0 xxxxxx	Write Parameter, reception
Read Parameter, reception	54	1011	0000		10	xxxxxx	xxxxxxxx		xxxxxxxx	xxxxxxxx	0 0 xxxxxx	Read Parameter, transmission
Idle_1	55	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	56	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	57	1111	0001		10	xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1

Figure I.1 – Exemple de transmissions d'ISDU *(1 de 2)*

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

ISDURequest_2, transmission	58	0111 0000	10 xxxxxx	xxxxxxxx	0001 1011		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	59	0110 0001	10 xxxxxx	xxxxxxxx	Index		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	60	0110 0010	10 xxxxxx	xxxxxxxx	Data 1		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	61	0110 0011	10 xxxxxx	xxxxxxxx	Data 2		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	62	0110 0100	10 xxxxxx	xxxxxxxx	Data 3		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	63	0110 0101	10 xxxxxx	xxxxxxxx	Data 4		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	64	0110 0110	10 xxxxxx	xxxxxxxx	Data 5		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	65	0110 0111	10 xxxxxx	xxxxxxxx	Data 6		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	66	0110 1000	10 xxxxxx	xxxxxxxx	Data 7		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	67	0110 1001	10 xxxxxx	xxxxxxxx	Data 8		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	68	0110 1010	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	69	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	70	1111 0000	10 xxxxxx	xxxxxxxx		0101 0010	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	71	1110 0001	10 xxxxxx	xxxxxxxx		CHKPDU	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
Idle_1	72	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	73	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
ISDURequest_2, transmission,	74	0111 0000	10 xxxxxx	xxxxxxxx	1011 0101		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	75	0110 0001	10 xxxxxx	xxxxxxxx	Index(hi)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	76	0110 0010	10 xxxxxx	xxxxxxxx	Index(lo)		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	77	0110 0011	10 xxxxxx	xxxxxxxx	Subindex		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDURequest_2, transmission	78	0110 0100	10 xxxxxx	xxxxxxxx	CHKPDU		xxxxxxxx	0 0 xxxxxx	ISDURequest_2, reception
ISDUWait_3, start ISDU Timer	79	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	80	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	81	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	82	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUWait_3, inc. ISDU timer	83	1111 0000	10 xxxxxx	xxxxxxxx		0000 0001	xxxxxxxx	0 0 xxxxxx	ISDUWait_3, application busy
ISDUResponse_4, reception Stop ISDU Timer	84	1111 0000	10 xxxxxx	xxxxxxxx		1101 0001	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	85	1110 0001	10 xxxxxx	xxxxxxxx		0001 1110	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, reception	86	1110 0010	10 xxxxxx	xxxxxxxx		Data 1	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, transmission
ISDUResponse_4, ABORT	87	1111 1111	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	ISDUResponse_4, ABORT
Idle_1	88	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1
Idle_1	89	1111 0001	10 xxxxxx	xxxxxxxx		0000 0000	xxxxxxxx	0 0 xxxxxx	Idle_1

Figure I.1 *(2 de 2)*

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Annexe J
(informative)

Méthodes recommandées pour la détection de modifications de paramètres

J.1 Signature CRC

Le contrôle de redondance cyclique fait partie de la famille des fonctions de HASH. Une signature CRC dans tous les paramètres modifiables peut être calculée par le Dispositif au moyen de ce que l'on appelle un polynôme générateur correct. Le calcul donne une signature différente à chaque fois que le jeu de paramètres est modifié. Il convient de noter que la signature protège également l'ordre des octets dans le jeu de paramètres. Toute modification de cet ordre observée au moment où la signature est calculée donnera une valeur différente. La qualité de la protection (modifications non détectées) dépend fortement à la fois du polynôme générateur de CRC et de la longueur (nombre d'octets) du jeu de paramètres. Il convient que la valeur de départ soit > 0. L'une des méthodes de calcul utilise directement la formule, tandis que l'autre utilise des tables de décalage et de consultation d'octets. La première, qui est légèrement plus lente, nécessite une moindre mémoire programme, tandis que l'autre, rapide, nécessite de la mémoire pour la table de consultation (1×2^{10} octets pour une signature de 32 bits). La comparaison des jeux de données de paramètres est effectuée à l'état "Checksum_9" du diagramme d'états du Stockage des Données (DS), comme illustré en Figure 100. Le Tableau J.1 énumère plusieurs polynômes générateurs possibles et leurs niveaux de protection.

Tableau J.1 – Polynômes générateurs de CRC corrects

Polynôme générateur	Signature	Longueur de données	Modifications non détectées
0x9B	8 bits	1 octet	$< 2^{-8}$ (non recommandé)
0x4EAB	16 bits	1 < octets < 3	$< 2^{-16}$ (non recommandé)
0x5D6DCB	24 bits	1 < octets < 4	$< 2^{-24}$ (non recommandé)
0xF4ACFB13	32 bits	1 < octets < 2^{32}	$< 2^{-32}$ (recommandé)

J.2 Compteur de versions

Il peut être utilisé un compteur de versions de 32 bits, qui compte toute modification du jeu de paramètres. Le Dispositif doit utiliser une valeur initiale aléatoire pour le compteur de versions. Le compteur proprement dit ne doit pas être enregistré via la liste d'Index du Dispositif. Après téléchargement aval, la valeur réelle du compteur est relue par le Dispositif pour éviter des écritures multiples induites par la séquence de téléchargement. La comparaison des jeux de données de paramètres est effectuée à l'état "Checksum_9" du diagramme d'états du Stockage des Données (DS), comme illustré en Figure 100.

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Bibliographie

[1] CEI 60050 (toutes parties), *Vocabulaire Électrotechnique International* (disponible à l'adresse <http://www.electropedia.org>)

[2] CEI 60870-5-1:1990, *Matériels et systèmes de téléconduite – Cinquième Partie: Protocoles de transmission – Section Un: Formats de trames de transmission*

[3] CEI 61158-2, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 2: Spécification de couche physique et définition des services*

[4] CEI/TR 62453-61, *Field device tool interface specification – Part 61:Device Type Manager (DTM) Styleguide for common object model* (disponible en anglais seulement)

[5] ISO/CEI 7498-1, *Technologie de l'information – Modèle de référence de base pour l'interconnexion de systèmes ouverts (OSI): Le modèle de base*

[6] IO-Link Consortium, *IO Device Description (IODD), V1.1, Order No. 10.012,* disponible à l'adresse <http://www.io-link.com> (disponible en anglais seulement)

[7] IO-Link Consortium, *IO-Link Smart Sensor Profile, V1.1, Order No. 10.042,* disponible à l'adresse <http://www.io-link.com> (disponible en anglais seulement)

[8] IO-Link Consortium, *IO-Link Communication, V1.0, January 2009, Order No. 10.002,* disponible à l'adresse <http://www.io-link.com> (disponible en anglais seulement)

[9] IO-Link Consortium, *IO-Link Test Specification, V1.1, Order No. 10.032,* disponible à l'adresse <http://www.io-link.com> (disponible en anglais seulement)

_____

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when print

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when print

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch

Copyrighted material licensed to BR Demo by Thomson Reuters (Scientific), Inc., subscriptions.techstreet.com, downloaded on Nov-27-2014 by James Madison. No further reproduction or distribution is permitted. Uncontrolled when printe