

# INTERNATIONAL STANDARD

**IEC**  
**61158-6**

Third edition  
2003-05

---

---

**Digital data communications  
for measurement and control –  
Fieldbus for use in industrial  
control systems –**

**Part 6:  
Application layer protocol specification**

*(https://standards.iteh.ai)*  
*Document Preview*

[IEC 61158-6:2003](https://standards.iteh.ai/standards/iec/61158-6:2003)

<https://standards.iteh.ai/standards/iec/61158-6:2003>



Reference number  
IEC 61158-6:2003(E)

## Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

## Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

## Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site** ([www.iec.ch](http://www.iec.ch))

- **Catalogue of IEC publications**

The on-line catalogue on the IEC web site ([http://www.iec.ch/searchpub/cur\\_fut.htm](http://www.iec.ch/searchpub/cur_fut.htm)) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

This summary of recently issued publications ([http://www.iec.ch/online\\_news/justpub/jp\\_entry.htm](http://www.iec.ch/online_news/justpub/jp_entry.htm)) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: [custserv@iec.ch](mailto:custserv@iec.ch)  
Tel: +41 22 919 02 11  
Fax: +41 22 919 03 00

# INTERNATIONAL STANDARD

# IEC 61158-6

Third edition  
2003-05

---

---

**Digital data communications  
for measurement and control –  
Fieldbus for use in industrial  
control systems –**

**Part 6:  
Application layer protocol specification**

*(https://standards.iteh.ai)*  
*Document Preview*

*IEC 61158-6:2003*

*https://standards.iteh.ai/standards/iec/3e8280c8-862e-412c-aea7-ca2c570d95f/iec-61158-6-2003*

© IEC 2003 – Copyright - all rights reserved

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 the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland  
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch



Commission Electrotechnique Internationale  
International Electrotechnical Commission  
Международная Электротехническая Комиссия

PRICE CODE

**XH**

*For price, see current catalogue*

CONTENTS

|  |     |
|--|-----|
| FOREWORD.....  | 32  |
| 0 Introduction.....  | 34  |
| 1 Scope .....  | 36  |
| 2 Normative references.....                                  | 37  |
| 3 Terms and definitions.....                                 | 38  |
| 3.1 Summary.....   | 38  |
| 3.2 Terms and definitions from other ISO/IEC standards ..... | 38  |
| 3.3 Terms and definitions from IEC 61158-5.....              | 39  |
| 3.4 Other terms and definitions.....                         | 39  |
| 3.5 Abbreviations and symbols.....                           | 49  |
| 3.6 Conventions .....  | 53  |
| 3.7 Conventions used in state machines.....                  | 61  |
| 4 Type 1 .....   | 68  |
| 4.1 FAL syntax description .....                             | 68  |
| 4.2 Transfer syntaxes.....                                   | 107 |
| 4.3 FAL protocol state machines structure.....               | 132 |
| 4.4 AP-Context state machine .....                           | 134 |
| 4.5 FAL Service Protocol Machine (FSPM).....                 | 151 |
| 4.6 Application Relationship Protocol Machines (ARPMs).....  | 157 |
| 4.7 DLL Mapping Protocol Machine (DMPM).....                 | 298 |
| 4.8 Protocol options .....                                   | 312 |
| 5 Type 2 .....   | 316 |
| 5.1 Abstract syntax.....                                     | 316 |
| 5.2 Transfer syntax .....                                    | 390 |
| 5.3 Structure of FAL protocol state machines .....           | 402 |
| 5.4 Context state machine.....                               | 402 |
| 5.5 FAL Service Protocol Machine (FSPM).....                 | 402 |
| 5.6 Application Relationship Protocol Machines (ARPMs).....  | 408 |
| 5.7 DLL Mapping Protocol Machine (DMPM) .....                | 492 |
| 5.8 Alternate DLL mapping protocol machine.....              | 496 |
| 6 Type 3 .....   | 502 |
| 6.1 FAL syntax description .....                             | 502 |
| 6.2 Transfer syntax .....                                    | 506 |
| 6.3 FAL protocol state machines .....                        | 553 |
| 6.4 AP-Context state machine .....                           | 567 |
| 6.5 FAL Service Protocol Machines (FSPMs) .....              | 567 |
| 6.6 Application Relationship Protocol Machines (ARPMs).....  | 650 |
| 6.7 DLL Mapping Protocol Machines (DMPMs).....               | 791 |
| 6.8 Parameters for a DP-slave.....                           | 827 |
| 7 Type 4 .....   | 828 |
| 7.1 FAL syntax description .....                             | 828 |
| 7.2 Transfer syntaxes.....                                   | 830 |
| 7.3 FAL Protocol State Machines.....                         | 837 |
| 7.4 AP-Context State Machine.....                            | 838 |
| 7.5 FAL Service Protocol Machine (FSPM).....                 | 838 |

|      |  |      |
|------|--|------|
| 7.6  | Application Relationship Protocol Machine (ARPM) .....   | 843  |
| 7.7  | DLL Mapping Protocol Machine (DMPM) .....  | 846  |
| 7.8  | Protocol options .....   | 849  |
| 8    | Type 5 .....   | 850  |
| 8.1  | Overview .....   | 850  |
| 8.2  | FAL Syntax Description .....   | 850  |
| 8.3  | Transfer syntax .....  | 850  |
| 8.4  | FAL Protocol State Machine Structure .....   | 899  |
| 8.5  | SMK State Machine .....  | 899  |
| 8.6  | VCR State Machine .....  | 914  |
| 8.7  | FAL Service Protocol Machine (FSPM) .....  | 915  |
| 8.8  | Application Relationship Protocol Machines (ARPMs) .....                                       | 916  |
| 8.9  | DLL Mapping Protocol Machine (DMPM) .....  | 930  |
| 9    | Type 6 .....   | 936  |
| 9.1  | Reference .....  | 936  |
| 9.2  | Usage of Parameters .....  | 936  |
| 9.3  | AP Context State Tables .....  | 936  |
| 9.4  | Application Relationship protocol machines (ARPMs) .....                                       | 936  |
| 9.5  | DLL Mapping protocol machine (DMPM) .....  | 952  |
| 10   | Type 7 .....   | 962  |
| 10.1 | Abstract syntax of data type .....   | 962  |
| 10.2 | Transfer Syntaxes .....  | 967  |
| 10.3 | Structure of Protocol Machines .....   | 1027 |
| 10.4 | AP-Context state machine .....   | 1028 |
| 10.5 | Sub-MMS FAL Service Protocol Machine (FSPM) .....  | 1028 |
| 10.6 | DLL Mapping Protocol Machine (DMPM) and Association Relationship Protocol Machine (ARPM) ..... | 1033 |
| 10.7 | Protocol options .....   | 1081 |
| 11   | Type 8 .....   | 1101 |
| 11.1 | FAL Syntax Description .....   | 1101 |
| 11.2 | Transfer Syntax .....  | 1110 |
| 11.3 | Protocol Machine Overview .....  | 1118 |
| 11.4 | AP-Context State Machine .....   | 1119 |
| 11.5 | FAL Service Protocol Machine (FSPM) .....  | 1119 |
| 11.6 | Application Relationship Protocol Machines (ARPMs) .....                                       | 1119 |
| 11.7 | DLL Mapping Protocol Machine .....   | 1128 |
| 12   | Type 9 .....   | 1139 |
| 12.1 | FAL syntax description .....   | 1139 |
| 12.2 | Transfer Syntax .....  | 1158 |
| 12.3 | FAL Protocol State Machines Structure .....  | 1167 |
| 12.4 | VCR PM State Machine .....   | 1170 |
| 12.5 | AREP State Machines .....  | 1182 |
| 13   | Type 10 .....  | 1230 |
| 13.1 | FAL Syntax Description .....   | 1230 |
| 13.2 | Transfer Syntax .....  | 1247 |
| 13.3 | FAL Protocol State Machines .....  | 1247 |

|   |      |
|---|------|
| 13.4 AP Context State Machine .....                         | 1248 |
| 13.5 FAL Service Protocol Machines (FSPM) .....             | 1248 |
| 13.6 Application Relationship Protocol Machine (ARPM) ..... | 1303 |
| 13.7 DLL Mapping Protocol Machine (DMPM) .....              | 1306 |
| 13.8 Protocol Options .....                                 | 1309 |

|   |     |
|---|-----|
| Figure 1 – Relationship of IEC 61158-6 to other fieldbus layers and to users of the fieldbus Application service..... | 34  |
| Figure 2 – Attribute table format and terms .....   | 53  |
| Figure 3 – Service request/response parameter .....   | 54  |
| Figure 4 – Common structure of specific fields.....   | 58  |
| Figure 5 – Example of an STD .....  | 63  |
| Figure 6 – Example of an evaluation net .....   | 67  |
| Figure 7 – APDU overview .....  | 108 |
| Figure 8 – Identification information (format 1).....   | 108 |
| Figure 9 – Identification information (format 2).....   | 109 |
| Figure 10 – Identification information (format 3).....  | 109 |
| Figure 11 – Identification information (format 4).....  | 109 |
| Figure 12 – Coding of the data type BinaryDate .....  | 111 |
| Figure 13 – Encoding of Time of Day value.....  | 111 |
| Figure 14 – Encoding of Time Difference value.....  | 112 |
| Figure 15 – Encoding Time value .....   | 112 |
| Figure 16 – Encoding of data of data type DL—Time-offset.....   | 113 |
| Figure 17 – Structure of an object definition.....  | 114 |
| Figure 18 – APDU overview.....  | 117 |
| Figure 19 – Type field .....  | 117 |
| Figure 20 – Identifier octet (context-specific) .....   | 118 |
| Figure 21 – Identifier octet (FAL-specific) .....   | 118 |
| Figure 22 – Length octet (one-octet format) .....   | 119 |
| Figure 23 – Length octet (three-octet format).....  | 119 |
| Figure 24 – APDU overview .....   | 127 |
| Figure 25 – Relationships among protocol machines and adjacent layers .....   | 133 |
| Figure 26 – AP to AP-Context initiation state machine .....   | 135 |
| Figure 27 – State transition diagram of FSPM.....   | 153 |
| Figure 28 – State transition diagram of the QUU ARPM .....  | 160 |
| Figure 29 – State transition diagram of QUB-CO ARPM.....  | 169 |
| Figure 30 – State transition diagram of the QUB-CL ARPM.....  | 181 |
| Figure 31 – State transition diagram of QUB-seg ARPM .....  | 191 |
| Figure 32 – State transition diagram of QUB-FC ARPM .....   | 211 |
| Figure 33 – State transition diagram of BUB ARPM.....   | 238 |
| Figure 34 – State transition diagram of BNB ARPM (basic state machine) .....  | 254 |
| Figure 35 – State transition diagram of BNB ARPM (confirmed service sending and receiving – client) .....             | 255 |
| Figure 36 – State transition diagram of BNB ARPM (confirmed service receiving and responding – server).....           | 255 |
| Figure 37 – State transition diagram of BNB ARPM (unconfirmed service sending- client) ..                             | 255 |
| Figure 38 – State transition diagram of the BNU ARPM .....  | 279 |

|  |     |
|--|-----|
| Figure 39 – State transition diagram of the BNU-MP ARPM .....  | 291 |
| Figure 40 – State transition diagram of DMPM .....   | 304 |
| Figure 41 – Primitives exchanged between protocol machines .....   | 313 |
| Figure 42 – Primitives exchanged between protocol machines .....   | 314 |
| Figure 43 – Primitives exchanged between protocol machines .....   | 315 |
| Figure 44 – Network connection parameters .....  | 333 |
| Figure 45 – Time tick .....  | 335 |
| Figure 46 – Connection establishment time-out .....  | 337 |
| Figure 47 – Segment type .....   | 352 |
| Figure 48 – Port segment .....   | 353 |
| Figure 49 – Encapsulation message .....  | 377 |
| Figure 50 – FixedLengthBitString compact encoding bit placement rules .....  | 394 |
| Figure 51 – Example compact encoding of a OCTET FixedLengthBitString .....   | 394 |
| Figure 52 – Example compact encoding of a WORD FixedLengthBitString .....  | 395 |
| Figure 53 – Example compact encoding of a DWORD FixedLengthBitString .....   | 395 |
| Figure 54 – Example compact encoding of a LWORD FixedLengthBitString .....   | 395 |
| Figure 55 – Example 2 of formal encoding of a structure type specification .....   | 399 |
| Figure 56 – Example of abbreviated encoding of a structure type specification .....  | 399 |
| Figure 57 – Example 1 of formal encoding of an array type specification .....  | 400 |
| Figure 58 – Example 2 of formal encoding of an array type specification .....  | 401 |
| Figure 59 – Example 1 of abbreviated encoding of an array type specification .....   | 401 |
| Figure 60 – Example 2 of abbreviated encoding of an array type specification .....   | 402 |
| Figure 61 – State transition diagram of UCMM client .....  | 411 |
| Figure 62 – State transition diagram of high-end UCMM server .....   | 413 |
| Figure 63 – State transition diagram of low-end UCMM server .....  | 415 |
| Figure 64 – Sequence diagram for a UCMM with one outstanding message .....   | 416 |
| Figure 65 – Sequence diagram for a UCMM with multiple outstanding messages .....   | 417 |
| Figure 66 – TPDU buffer .....  | 418 |
| Figure 67 – Data flow diagram using a client transport class 0 and server transport class 0 .....  | 420 |
| Figure 68 – Sequence diagram of data transfer using transport class 0 .....  | 421 |
| Figure 69 – Class 0 client STD .....   | 422 |
| Figure 70 – Class 0 server STD .....   | 423 |
| Figure 71 – Data flow diagram using client transport class 1 and server transport class 1 .....  | 424 |
| Figure 72 – Sequence diagram of data transfer using client transport class 1 and server transport class 1 .....                              | 425 |
| Figure 73 – Class 1 client STD .....   | 427 |
| Figure 74 – Class 1 server STD .....   | 428 |
| Figure 75 – Data flow diagram using client transport class 2 and server transport class 2 .....  | 430 |
| Figure 76 – Diagram of data transfer using client transport class 2 and server transport class 2 <i>without</i> returned data .....          | 431 |
| Figure 77 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 <i>with</i> returned data .....    | 432 |
| Figure 78 – Class 2 client STD .....   | 434 |
| Figure 79 – Class 2 server STD .....   | 436 |
| Figure 80 – Data flow diagram using client transport class 3 and server transport class 3 .....  | 439 |
| Figure 81 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 <i>without</i> returned data ..... | 440 |

Figure 82 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 *with* returned data ..... 441

Figure 83 – Class 3 client STD..... 443

Figure 84 – Class 3 server STD ..... 445

Figure 85 – Data flow diagram using transport classes 4 and 5..... 447

Figure 86 – Sequence diagram of message exchange using transport classes 4 and 5 ..... 448

Figure 87 – Sequence diagram of messages overwriting each other ..... 449

Figure 88 – Sequence diagram of queued message exchange using transport classes 4 and 5 ..... 450

Figure 89 – Sequence diagram of retries using transport classes 4 and 5 ..... 451

Figure 90 – Sequence diagram of idle traffic using transport classes 4 and 5..... 452

Figure 91 – Classes 4 and 5 basic structure..... 453

Figure 92 – Class 6 basic structure ..... 454

Figure 93 – Classes 4 to 6 general STD..... 455

Figure 94 – Class 4 sender STD ..... 457

Figure 95 – Class 4 receiver STD ..... 460

Figure 96 – Sequence diagram of three fragments using transport class 5 ..... 463

Figure 97 – Sequence diagram of fragmentation with retries using transport class 5 ..... 464

Figure 98 – Sequence diagram of two fragments using transport class 5 ..... 465

Figure 99 – Sequence diagram of aborted message using transport class 5..... 465

Figure 100 – Class 5 sender STD ..... 467

Figure 101 – Class 5 receiver STD ..... 470

Figure 102 – Data flow diagram for transport class 6 ..... 474

Figure 103 – Sequence diagram of message exchange using transport class 6 ..... 476

Figure 104 – Sequence diagram of retries using transport class 6..... 476

Figure 105 – Sequence diagram of idle traffic using transport class 6 ..... 477

Figure 106 – Sequence diagram of request overwriting null ..... 478

Figure 107 – Sequence diagram of response overwriting ACK of null..... 479

Figure 108 – Sequence diagram of three fragments using transport class 6 ..... 480

Figure 109 – Sequence diagram of fragmentation with retries using transport class 6 ..... 481

Figure 110 – Sequence diagram of two fragments using transport class 6..... 482

Figure 111 – Sequence diagram of aborted fragmented sequence using transport class 6 .. 483

Figure 112 – Class 6 client STD..... 484

Figure 113 – Class 6 server STD ..... 487

Figure 114 – Data flow diagram for a link producer and consumer ..... 492

Figure 115 – State transition diagram for a link producer ..... 495

Figure 116 – State transition diagram for a link consumer..... 496

Figure 117 – Coding of the data type BinaryDate ..... 508

Figure 118 – Encoding of Time Of Day value ..... 508

Figure 119 – Encoding of Time Difference value ..... 508

Figure 120 – Encoding of Network Time value ..... 509

Figure 121 – Encoding of Network Time Difference value ..... 509

Figure 122 – Example Modul\_Status\_Array..... 514

Figure 123 – Example of Ext\_Diag\_Data in case of DPV1 diagnosis format with alarm and status PDU..... 549

Figure 124 – Example of Ext\_Diag\_Data in case of the basic diagnosis format ..... 550

Figure 125 – Example of a special identifier format..... 550



|   |     |
|---|-----|
| Figure 126 – Example of a special identifier format with data types.....                                | 551 |
| Figure 127 – Example of a special identifier format with data types.....                                | 551 |
| Figure 128 – Example of a empty slot with data types.....   | 551 |
| Figure 129 – Example for multi-variable device with AI and DO function blocks .....                     | 552 |
| Figure 130 – Identifiers (ID).....  | 552 |
| Figure 131 – Identifier list .....  | 552 |
| Figure 132 – Structure of the Data_Unit for the request- and response-DLPDU .....                       | 553 |
| Figure 133 – Structuring of the protocol machines and adjacent layers in a DP-slave .....               | 556 |
| Figure 134 – Structuring of the protocol machines and adjacent layers in a DP-master<br>(class 1) ..... | 557 |
| Figure 135 – Structuring of the protocol machines and adjacent layers in a DP-master<br>(class 2) ..... | 558 |
| Figure 136 – Sequence of the communication between DP-master and DP-slave .....                         | 560 |
| Figure 137 – Sequence of communication between DP-master (class 2) and DP-master<br>(class 1) .....     | 562 |
| Figure 138 – Sequence of acyclic communication between DP-master (class 1)<br>and DP-slave.....         | 563 |
| Figure 139 – Example for connection establishment on MS2.....   | 565 |
| Figure 140 – Idle at master-side on MS2.....  | 566 |
| Figure 141 – Idle at slave-side on MS2.....   | 567 |
| Figure 142 – Example for connection establishment on MS2(server-side) .....                             | 696 |
| Figure 143 – Structure of RM entries in the RM_Registry.....  | 697 |
| Figure 144 – APDU Header structure.....   | 830 |
| Figure 145 – Instruction subfield of ControlStatus.....   | 830 |
| Figure 146 – Errorcode subfield of ControlStatus.....   | 831 |
| Figure 147 – Remaining subfields of ControlStatus.....  | 831 |
| Figure 148 – DataFieldFormat encoding.....  | 832 |
| Figure 149 – Structure of request APDU Body .....   | 832 |
| Figure 150 – Structure of response APDU Body.....   | 832 |
| Figure 151 – Variable Identifier.....   | 832 |
| Figure 152 – Code subfield of Variable Identifier.....  | 833 |
| Figure 153 – Summary of FAL Architecture.....   | 837 |
| Figure 154 – FSPM proxy object state machine .....  | 839 |
| Figure 155 – FSPM real object state machine .....   | 843 |
| Figure 156 – ARPM State Machine .....   | 844 |
| Figure 157 – DLPM State Machine.....  | 847 |
| Figure 158 – State Transition Diagram for SMK .....   | 901 |
| Figure 159 – State Transition Diagram of Client / Server ARPM.....                                      | 919 |
| Figure 160 – State Transition diagram of the publisher / subscriber ARPM .....                          | 926 |
| Figure 161 – State transition diagram of DMPM .....   | 932 |
| Figure 162 – State transition diagram of QUB-PC ARPM .....  | 940 |
| Figure 163 – State transition diagram of the BNU-PC ARPM.....   | 950 |
| Figure 164 – State transition diagram of DMPM .....   | 955 |
| Figure 165 – Encoding of a CompactValue .....   | 967 |
| Figure 166 – Organisation of the bits and octets within a PDU .....                                     | 968 |
| Figure 167 – Encoding of a Bitstring .....  | 972 |
| Figure 168 – Encoding of a Floating point.....  | 973 |

|  |      |
|--|------|
| Figure 169 – Encoding of a structure .....   | 974  |
| Figure 170 – Encoding of a boolean array.....  | 975  |
| Figure 171 – Representation of a MCS PDU .....   | 981  |
| Figure 172 – Relationships among Protocol Machines and Adjacent Layers.....              | 1027 |
| Figure 173 – A_Readloc service evaluation net.....                                       | 1033 |
| Figure 174 – A_Writeloc service evaluation net.....                                      | 1034 |
| Figure 175 – A_Update service evaluation net .....                                       | 1035 |
| Figure 176 – A_Readfar service evaluation net.....                                       | 1037 |
| Figure 177 – A_writefar service evaluation net.....                                      | 1039 |
| Figure 178 – A_Sent service evaluation net.....  | 1040 |
| Figure 179 – A_Received service evaluation net.....                                      | 1040 |
| Figure 180 – Association establishment: Requester element state machine.....             | 1047 |
| Figure 181 – Association establishment: Responder element state machine.....             | 1048 |
| Figure 182 – Association termination: Requester element state machine.....               | 1050 |
| Figure 183 – Association termination: Responder element state machine.....               | 1052 |
| Figure 184 – Association revocation: Requester element state machine.....                | 1053 |
| Figure 185 – Association revocation: Acceptor element state machine.....                 | 1054 |
| Figure 186 – Interactions between state machine in an associated mode data transfer..... | 1056 |
| Figure 187 – Transfer service – Requester element state machine.....                     | 1060 |
| Figure 188 – Transfer service: Acceptor element state machine.....                       | 1061 |
| Figure 189 – Unacknowledged transfer: Requester element state machine.....               | 1062 |
| Figure 190 – Unacknowledged transfer: Acceptor element state machine.....                | 1062 |
| Figure 191 – Acknowledged transfer: Requester element state machine.....                 | 1064 |
| Figure 192 – Acknowledged transfer: Acceptor element state machine.....                  | 1065 |
| Figure 193 – Numbering mechanism state machine.....                                      | 1066 |
| Figure 194 – Retry mechanism state machine.....  | 1068 |
| Figure 195 – Anticipation mechanism state machine.....                                   | 1071 |
| Figure 196 – Segmentation mechanism state machine.....                                   | 1073 |
| Figure 197 – Reassembly mechanism state machine.....                                     | 1075 |
| Figure 198 – Interaction of state machine in a non associated data transfer.....         | 1077 |
| Figure 199 – Unacknowledged transfer: Requester element state machine.....               | 1078 |
| Figure 200 – Unacknowledged transfer: Acceptor element state machine.....                | 1078 |
| Figure 201 – Acknowledged transfer: Requester element state machine.....                 | 1080 |
| Figure 202 – Acknowledged transfer: Acceptor element state machine.....                  | 1081 |
| Figure 203 – APDU overview .....   | 1111 |
| Figure 204 – APDU Header.....  | 1111 |
| Figure 205 – PDU withType Extension.....   | 1111 |
| Figure 206 – PDU with Address Extension.....   | 1111 |
| Figure 207 – PDU with Type and Length Extension.....                                     | 1111 |
| Figure 208 – Example of an Establish-Request PDU.....                                    | 1112 |
| Figure 209 – Encoding of a PRIVATE tagged value .....                                    | 1112 |
| Figure 210 – Encoding of a context specific tagged value .....                           | 1113 |
| Figure 211 – Identification information fields.....                                      | 1113 |
| Figure 212 – ID-info for Tag 0 .. 14 , Length entry 0 .. 6.....                          | 1113 |
| Figure 214 – ID-info for Tag 15 .. 255 , Length entry 0 .. 6.....                        | 1113 |
| Figure 215 – ID-info for Tag 0 .. 14 , Length entry 7 .. 255.....                        | 1113 |

|   |      |
|---|------|
| Figure 216 – ID-info for Tag 15 .. 255 , Length entry 7 .. 255 .....                  | 1114 |
| Figure 217 – Encoding of Boolean value TRUE.....                                      | 1114 |
| Figure 218 – Encoding of Boolean value FALSE .....                                    | 1114 |
| Figure 219 – Encoding of Strings .....  | 1114 |
| Figure 220 – Encoding of BinaryDate Value .....                                       | 1115 |
| Figure 221 – Encoding of BinaryDate2000 Value .....                                   | 1115 |
| Figure 222 – Encoding of Time of Day value .....                                      | 1116 |
| Figure 223 – Encoding of Time Difference Value.....                                   | 1116 |
| Figure 224 – Encoding of Time Value .....   | 1117 |
| Figure 225 – Example for an Object Definition .....                                   | 1118 |
| Figure 226 – Primitives Exchanged between Protocol Machines .....                     | 1119 |
| Figure 227 – State Transition Diagram of QUB-TM AREP.....                             | 1126 |
| Figure 228 – State transition diagram of Type-8 DMPM .....                            | 1131 |
| Figure 229 – Structure of an object description in the OD .....                       | 1154 |
| Figure 230 – Structure of the OD object description .....                             | 1154 |
| Figure 231 – Structure of a domain entry in the S-OD .....                            | 1154 |
| Figure 232 – Object description of program invocation in the DP-OD .....              | 1154 |
| Figure 233 – Object description of simple variable in the S-OD .....                  | 1154 |
| Figure 234 – Object description of array in the S-OD .....                            | 1154 |
| Figure 235 – Object description of record in the S-OD .....                           | 1155 |
| Figure 236 – Object description of variable list in the DV-OD .....                   | 1155 |
| Figure 237 – Object description of data type in the ST-OD.....                        | 1155 |
| Figure 238 – Object description of data type structure description in the ST-OD ..... | 1155 |
| Figure 239 – Object description of event in the S-OD.....                             | 1155 |
| Figure 240 – Insertion of identification information in the FMS PDU .....             | 1158 |
| Figure 241 – Coding of the ID info with and without extension .....                   | 1160 |
| Figure 242 – Coding with identification .....   | 1160 |
| Figure 243 – Coding without identification.....                                       | 1160 |
| Figure 244 – Representation of the value true.....                                    | 1160 |
| Figure 245 – Representation of the value false .....                                  | 1160 |
| Figure 246 – Coding of data of data type Integer16.....                               | 1161 |
| Figure 247 – Coding of data of data type Unsigned16 .....                             | 1161 |
| Figure 248 – Coding of data of data type Floating Point.....                          | 1162 |
| Figure 249 – Coding of data of data type Visible String.....                          | 1162 |
| Figure 250 – Coding of data of data type Octet String .....                           | 1162 |
| Figure 251 – Coding of data of data type Date .....                                   | 1163 |
| Figure 252 – Coding of data of data type Time Of Day.....                             | 1164 |
| Figure 253 – Coding of data of data type Time Difference.....                         | 1164 |
| Figure 254 – Coding of data of data type Bit String .....                             | 1165 |
| Figure 255 – Coding of data of data type Time Value.....                              | 1165 |
| Figure 256 – Coding of data of user data definitions with identifier.....             | 1165 |
| Figure 257 – Coding of data of user data definitions without identifier.....          | 1166 |
| Figure 258 – Coding of ID info for a SEQUENCE .....                                   | 1166 |
| Figure 259 – Relationships among protocol machines and adjacent layers .....          | 1168 |
| Figure 260 – VCR state machine .....  | 1172 |
| Figure 261 – State transition diagram of FSPM .....                                   | 1193 |

|  |      |
|--|------|
| Figure 262 – State transition diagram of the QUU ARPM .....                                | 1196 |
| Figure 263 – State transition diagram of QUB ARPM .....                                    | 1199 |
| Figure 264 – State transition diagram of the BNU ARPM.....                                 | 1208 |
| Figure 265 – State transition diagram of DMPM .....  | 1213 |
| Figure 266 - Error Message structure.....  | 1230 |
| Figure 267 – Coding scheme of ITEMQUALITYDEF .....   | 1233 |
| Figure 268 – Relationship among Protocol Machines .....                                    | 1248 |
| Figure 269 – State Transition Diagram of FSPM .....  | 1258 |
| Figure 270 – State Transition Diagram of ARPM .....  | 1304 |
| Figure 271 – State Transition Diagram of DMPM .....  | 1308 |
|  |      |
| Table 1– Get_Attribute_All response service rules .....                                    | 55   |
| Table 2 – Example class level object/service specific reply data of Get_Attribute_All..... | 55   |
| Table 3 – Example Get_Attribute_All data array method .....                                | 55   |
| Table 4 – Set_Attribute_All request service rules .....                                    | 56   |
| Table 5 – Example Set_Attribute_All attribute ordering method.....                         | 56   |
| Table 6 – Example Set_Attribute_All data array method.....                                 | 57   |
| Table 7 – Conventions used for state machines .....  | 61   |
| Table 8 – State event matrix format .....  | 63   |
| Table 9 – Example state event matrix .....   | 64   |
| Table 10 – State machine description elements .....  | 64   |
| Table 11 – Description of state machine elements .....                                     | 65   |
| Table 12 – Conventions used in state machines.....   | 65   |
| Table 13 – Reason codes .....  | 106  |
| Table 14 – FAL header .....  | 107  |
| Table 15 – Identification octet classes .....  | 128  |
| Table 16 – Unit of measurement of contents length octets .....                             | 129  |
| Table 17 – Primitives issued by FAL-user to AP-Context.....                                | 134  |
| Table 18 – Primitives issued by AP-Context to FAL-user.....                                | 134  |
| Table 19 – AP-Context state machine sender transactions.....                               | 136  |
| Table 20 – AP-Context state machine receiver transactions.....                             | 140  |
| Table 21 – Function ResetArep.....   | 147  |
| Table 22 – Function ApContextTest .....  | 148  |
| Table 23 – Function ServicesSupportedTest.....   | 148  |
| Table 24 – Function ApExplicitConnection .....   | 148  |
| Table 25 – Function ImmediateAcknowledge .....   | 148  |
| Table 26 – Function ConfirmedServiceCheck.....   | 148  |
| Table 27 – Function UnconfirmedServiceCheck .....  | 148  |
| Table 28 – Function ArServiceCheck .....   | 148  |
| Table 29 – Function ArFspmService .....  | 149  |
| Table 30 – Function ArAcceeSupported .....   | 149  |
| Table 31 – Function MaxFalPduLengthTest .....  | 149  |
| Table 32 – Function NegotiateOutstandingServices .....                                     | 149  |
| Table 33 – Function RequestedServicesSupportedTest .....                                   | 149  |
| Table 34 – Function IndicatedServicesSupportedTest.....                                    | 150  |
| Table 35 – Function InvokeldExistent .....   | 150  |

|  |     |
|--|-----|
| Table 36 – Function SameService.....   | 150 |
| Table 37 – Primitives issued by AP-Context to FSPM .....                         | 151 |
| Table 38 – Primitives issued by FSPM to AP-Context .....                         | 152 |
| Table 39 – FSPM state table – Sender transactions .....                          | 153 |
| Table 40 – FSPM state table – receiver transactions.....                         | 155 |
| Table 41 – Function SelectArep .....   | 156 |
| Table 42– Primitives issued by FSPM to ARPM .....                                | 157 |
| Table 43 – Primitives issued by ARPM to FSPM .....                               | 157 |
| Table 44 – Parameters used with primitives exchanged between FSPM and ARPM ..... | 157 |
| Table 45 – QUU ARPM states.....  | 160 |
| Table 46 – QUU ARPM state table – sender transactions.....                       | 161 |
| Table 47 – QUU ARPM state table – receiver transactions.....                     | 162 |
| Table 48 – Function GetArepId ().....  | 163 |
| Table 49 – Function BuildFAL-PDU.....  | 163 |
| Table 50 – Function FAL_Pdu_Type .....   | 163 |
| Table 51 – Primitives issued by FSPM to ARPM .....                               | 163 |
| Table 52 – Primitives issued by ARPM to FSPM .....                               | 164 |
| Table 53 – Parameters used with primitives exchanged between FSPM and ARPM ..... | 164 |
| Table 54 – QUB-CO ARPM states.....   | 169 |
| Table 55 – QUB-CO ARPM state table – sender transactions .....                   | 169 |
| Table 56 – QUB-CO ARPM state table – receiver transactions.....                  | 171 |
| Table 57 – Function GetArepId ().....  | 177 |
| Table 58 – Function BuildFAL-PDU.....  | 177 |
| Table 59 – Function FAL_Pdu_Type.....  | 177 |
| Table 60 – Function AbortIdentifier .....  | 177 |
| Table 61 – Function AbortReason.....   | 177 |
| Table 62 – Function AbortDetail.....   | 178 |
| Table 63 – Primitives issued by FSPM to ARPM .....                               | 178 |
| Table 64 – Primitives issued by ARPM to FSPM .....                               | 178 |
| Table 65 – Parameters used with primitives exchanged between FSPM and ARPM ..... | 179 |
| Table 66 – QUB-CL ARPM states .....  | 181 |
| Table 67 – QUB-CL ARPM state table – sender transactions .....                   | 182 |
| Table 68 – QUB-CL ARPM state table – receiver transactions .....                 | 183 |
| Table 69 – Function GetArepId ().....  | 184 |
| Table 70 – Function BuildFAL-PDU.....  | 185 |
| Table 71 – Function FAL-Pdu_Type.....  | 185 |
| Table 72 – Primitives issued by FSPM to ARPM .....                               | 185 |
| Table 73 – Primitives issued by ARPM to FSPM .....                               | 186 |
| Table 74 – Parameters used with primitives exchanged between FSPM and ARPM ..... | 186 |
| Table 75 – QUB-seg ARPM states .....   | 191 |
| Table 76 – QUB-seg ARPM state table – sender transactions .....                  | 192 |
| Table 77 – ARPM state table – receiver transactions .....                        | 195 |
| Table 78 – Function GetArepId .....  | 203 |
| Table 79 – Function BuildFAL-PDU.....  | 203 |
| Table 80 – Function FAL_Pdu_Type .....   | 204 |
| Table 81 – Function AbortIdentifier .....  | 204 |

|  |     |
|--|-----|
| Table 82 – Function AbortReason .....  | 204 |
| Table 83 – Function AbortDetail .....  | 204 |
| Table 84 – Function BuildFAL-segment .....   | 204 |
| Table 85 – Function MoreFollows .....  | 204 |
| Table 86 – Function ResetIntermediatePDU .....                                     | 205 |
| Table 87 – Function AddSegment .....   | 205 |
| Table 88 – Function GetIntermediatePDU .....                                       | 205 |
| Table 89 – Primitives issued by FSPM to ARPM .....                                 | 205 |
| Table 90 – Primitives issued by ARPM to FSPM .....                                 | 206 |
| Table 91 – Parameters used with primitives exchanged between FSPM and ARPM .....   | 206 |
| Table 92 – QUB-FC ARPM states .....  | 210 |
| Table 93 – QUB-FC ARPM state table – sender transactions .....                     | 211 |
| Table 94 – QUB-FC ARPM state table – receiver transactions .....                   | 217 |
| Table 95 – Function GetArepld () .....   | 229 |
| Table 96 – Function BuildFAL-PDU .....   | 229 |
| Table 97 – Function FAL_Pdu_Type .....   | 229 |
| Table 98 – Function AREPContextCheck() .....                                       | 230 |
| Table 99 – Function AbortIdentifier .....  | 230 |
| Table 100 – Function AbortReason .....   | 230 |
| Table 101 – Function AbortDetail .....   | 230 |
| Table 102 – Function StartTimer .....  | 230 |
| Table 103 – Function StopTimer .....   | 231 |
| Table 104 – Function ResetCounters .....   | 231 |
| Table 105 – Function IncrementCounter .....  | 231 |
| Table 106 – Function DecrementCounter .....  | 231 |
| Table 107 – Function GetCounterValue .....   | 231 |
| Table 108 – Function Xon .....   | 231 |
| Table 109 – Function SetXonOff .....   | 231 |
| Table 110 – Primitives issued by FSPM to ARPM .....                                | 232 |
| Table 111 – Primitives issued by ARPM to FSPM .....                                | 233 |
| Table 112 – Parameters used with primitives exchanged between FSPM and ARPM .....  | 233 |
| Table 113 – BUB ARPM states .....  | 237 |
| Table 114 – BUB ARPM state table – sender transactions .....                       | 239 |
| Table 115 – BUB state table – receiver transactions .....                          | 241 |
| Table 116 – Function GetArepld () .....  | 248 |
| Table 117 – Function BuildFAL-PDU .....  | 248 |
| Table 118 – Function FAL_Pdu_Type .....  | 248 |
| Table 119 – Function AbortIdentifier .....   | 248 |
| Table 120 – Function AbortReason .....   | 248 |
| Table 121 – Function AbortDetail .....   | 248 |
| Table 122 – Primitives issued by FSPM to ARPM .....                                | 249 |
| Table 123 – Primitives issued by ARPM to FSPM .....                                | 249 |
| Table 124 – Parameters used with primitives exchanged between FSPM and ARPM .....  | 250 |
| Table 125 – BNB ARPM states .....  | 254 |
| Table 126 – BNB ARPM state table – sender transactions (basic state machine) ..... | 256 |
| Table 127 – BNB state table – receiver transactions (basic state machine) .....    | 259 |