

BS EN 61800-7-202:2016



BSI Standards Publication

Adjustable speed electrical power drive systems

Part 7-202: Generic interface and use of
profiles for power drive systems — Profile
type 2 specification

bsi.

...making excellence a habit.
AVA **Documents**
Document Sharing Hub

National foreword

This British Standard is the UK implementation of EN 61800-7-202:2016. It is identical to IEC 61800-7-202:2015. It supersedes BS EN 61800-7-202:2008, which will be withdrawn on 12 October 2018.

The UK participation in its preparation was entrusted to Technical Committee PEL/22, Power electronics.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2016.

Published by BSI Standards Limited 2016

ISBN 978 0 580 82129 5

ICS 29.200; 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2016.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61800-7-202

February 2016

ICS 29.200; 35.100.05

Supersedes EN 61800-7-202:2008

English Version

Adjustable speed electrical power drive systems -
Part 7-202: Generic interface and use of profiles for power drive
systems - Profile type 2 specification
(IEC 61800-7-202:2015)

Entraînements électriques de puissance à vitesse variable -
Partie 7-202: Interface générique et utilisation de profils
pour les entraînements électriques de puissance -
Spécification de profil de type 2
(IEC 61800-7-202:2015)

Elektrische Leistungsantriebssysteme mit einstellbarer
Drehzahl - Teil 7-202: Generisches Interface und Nutzung
von Profilen für Leistungsantriebssysteme (PDS) -
Spezifikation von Profil-Typ 2
(IEC 61800-7-202:2015)

This European Standard was approved by CENELEC on 2015-12-25. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

European foreword

The text of document 22G/308/FDIS, future edition 2 of IEC 61800-7-202, prepared by SC 22G "Adjustable speed electric drive systems incorporating semiconductor power converters" of IEC/TC 22 "Power electronic systems and equipment" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61800-7-202:2016.

The following dates are fixed:

- latest date by which the document has to be (dop) 2016-09-25
implemented at national level by
publication of an identical national
standard or by endorsement
- latest date by which the national (dow) 2018-12-25
standards conflicting with the
document have to be withdrawn

This document supersedes EN 61800-7-202:2008.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 61800-7-202:2015 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61131-3	NOTE	Harmonized as EN 61131-3.
IEC 61158 Series	NOTE	Harmonized as EN 61158 Series.
IEC 61158-2:2014	NOTE	Harmonized as EN 61158-2:2014 (not modified).
IEC 61158-3-2:2014	NOTE	Harmonized as EN 61158-3-2:2014 (not modified).
IEC 61499-1:2005	NOTE	Harmonized as EN 61499-1:2005 ¹⁾ (not modified).
IEC 61784-1:2014	NOTE	Harmonized as EN 61784-1:2014 (not modified).
IEC 61784-2:2014	NOTE	Harmonized as EN 61784-2:2014 (not modified).
IEC 61800 Series	NOTE	Harmonized as EN 61800 Series.
IEC 61800-7 Series	NOTE	Harmonized as EN 61800-7 Series.
IEC 61800-7-201	NOTE	Harmonized as EN 61800-7-201.
IEC 61800-7-203	NOTE	Harmonized as EN 61800-7-203.
IEC 61800-7-204	NOTE	Harmonized as EN 61800-7-204.
IEC 61800-7-301	NOTE	Harmonized as EN 61800-7-301.
IEC 61800-7-302	NOTE	Harmonized as EN 61800-7-302.
IEC 61800-7-303	NOTE	Harmonized as EN 61800-7-303.
IEC 61800-7-304	NOTE	Harmonized as EN 61800-7-304.
IEC 62026-3	NOTE	Harmonized as EN 62026-3.

¹⁾ Superseded by EN 61499-1:2013 (IEC 61499-1:2012).

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 60204-1	-	Safety of machinery - Electrical equipment of machines - Part 1: General requirements	EN 60204-1	-
IEC 61158-4-2	2014	Industrial communication networks - Fieldbus specifications - Part 4-2: Data-link layer protocol specification - Type 2 elements	EN 61158-4-2	2014
IEC 61158-5-2	2014	Industrial communication networks - Fieldbus specifications - Part 5-2: Application layer service definition - Type 2 elements	EN 61158-5-2	2014
IEC 61158-6-2	2014	Industrial communication networks - Fieldbus specifications - Part 6-2: Application layer protocol specification - Type 2 elements	EN 61158-6-2	2014
IEC 61588	2009	Precision clock synchronization protocol for networked measurement and control systems	-	-
IEC 61800-7-1	2015	Adjustable speed electrical power drive systems - Part 7-1: Generic interface and use of profiles for power drive systems - Interface definition	EN 61800-7-1	2016
IEEE Std 112	2004	IEEE Standard Test Procedure for Polyphase Induction Motors and Generators	-	-

CONTENTS

FOREWORD	10
INTRODUCTION	12
0.1 General	12
0.2 Patent declaration	15
1 Scope	17
2 Normative references	17
3 Terms, definitions and abbreviated terms	17
3.1 Terms and definitions	17
3.2 Abbreviated terms	25
4 Overview	25
4.1 General	25
4.2 Control modes	26
4.2.1 General	26
4.2.2 Control methods	26
4.2.3 Control nomenclature	27
4.2.4 Position control	27
4.2.5 Velocity control	28
4.2.6 Acceleration control	30
4.2.7 Torque control	30
4.2.8 No Control	31
5 Data types	32
5.1 Data type overview	32
5.2 Conventions	32
6 CIP Motion drive profile	32
6.1 Object model	32
6.1.1 Object overview	32
6.1.2 Object description	33
6.2 How objects affect behavior	34
6.3 Defining object interfaces	34
6.4 I/O connection messages	35
6.4.1 General	35
6.4.2 CIP Motion I/O Connection	35
6.4.3 Controller-to-Device Connection	39
6.4.4 Device-to-Controller Connection	59
6.4.5 Fixed Motion I/O connection format	68
6.4.6 CIP Motion I/O Connection timing model	69
6.5 Device startup procedure	85
6.5.1 General	85
6.5.2 Motion I/O Connection creation	85
6.5.3 Motion Device Axis Object configuration	88
6.5.4 Time Synchronization	90
6.6 Device visualisation	92
6.7 Ethernet/IP Quality of Service (QoS)	93
7 Motion Device Axis Object	93
7.1 General considerations	93

7.1.1	General	93
7.1.2	Revision history	93
7.1.3	Object overview	93
7.1.4	Motion Device Axis Object abstraction	94
7.1.5	Motion Control Axis Object	95
7.1.6	Device control classification.....	95
7.1.7	Required vs. Optional in implementation	96
7.2	Class attributes	107
7.2.1	General	107
7.2.2	Semantics.....	111
7.3	Instance attributes	114
7.3.1	General	114
7.3.2	Motion Control configuration attributes	116
7.3.3	Motion Scaling attributes	117
7.3.4	Connection Data attributes	123
7.3.5	Motor attributes	129
7.3.6	Feedback attributes	140
7.3.7	Event Capture attributes	149
7.3.8	Command reference generation attributes	155
7.3.9	Control mode attributes	158
7.3.10	Stopping & Braking attributes	176
7.3.11	DC Bus Control attributes	186
7.3.12	Power and thermal management attributes	190
7.3.13	Axis Status attributes.....	193
7.3.14	Exception, fault, and alarm attributes.....	199
7.3.15	Fault and alarm Log attributes	205
7.3.16	Exception limit attributes.....	210
7.3.17	Axis exception action configuration attribute	213
7.3.18	Initialization fault attributes	215
7.3.19	Start inhibit attributes	216
7.3.20	APR fault attributes	217
7.3.21	Axis statistical attributes	219
7.3.22	Axis info attributes	219
7.3.23	General purpose I/O attributes.....	220
7.3.24	Local Mode attributes	222
7.3.25	Axis Safety attributes.....	222
7.4	Common services	223
7.4.1	Supported services	223
7.4.2	Service specific data.....	224
7.5	Object specific services	226
7.5.1	Supported services	226
7.5.2	Service specific data.....	226
7.6	Behavior	241
7.6.1	State model	241
7.6.2	State behavior	252
7.6.3	Fault and alarm behavior	259
7.6.4	Start Inhibit behavior	261
7.6.5	Visualization behavior.....	261
7.6.6	Command generation behavior	

7.6.7	Feedback interface behavior.....	271
7.6.8	Event Capture Behavior.....	273
7.6.9	Control Mode behavior.....	274
Bibliography.....		288
Figure 1 – Structure of IEC 61800-7.....		15
Figure 2 – Open loop position control.....		27
Figure 3 – Closed loop position control		28
Figure 4 – Open loop velocity control		29
Figure 5 – Closed loop velocity control		29
Figure 6 – Acceleration control		30
Figure 7 – Torque control.....		31
Figure 8 – No Control (Feedback Only).....		31
Figure 9 – Object Model for a CIP Motion device		33
Figure 10 – CIP Motion I/O Connection model		35
Figure 11 – CIP Motion I/O Connection channels		36
Figure 12 – Controller-to-Device Connection format (Connection Point 2).....		37
Figure 13 – Device-to-Controller Connection format (Connection Point 2).....		38
Figure 14 – CIP Motion Controller-to-Device Connection format.....		39
Figure 15 – Connection Header		39
Figure 16 – Connection Format.....		39
Figure 17 – Connection Header		40
Figure 18 – Instance Data Block		43
Figure 19 – Instance Data Header		43
Figure 20 – Cyclic Data Block		44
Figure 21 – Control Mode		44
Figure 22 – Feedback Mode.....		44
Figure 23 – Cyclic Write Data Block		50
Figure 24 – Cyclic Write Data Block example		50
Figure 25 – Event Data Block.....		51
Figure 26 – Service Data Block.....		58
Figure 27 – CIP Motion Device-to-Controller Connection format.....		59
Figure 28 – Connection Header		59
Figure 29 – Connection Header		60
Figure 30 – Node Fault/Alarm		61
Figure 31 – Adjustment of actual position data based on Device Time Stamp		62
Figure 32 – Instance Data Block		63
Figure 33 – Instance Data Header		63
Figure 34 – Cyclic Data Block		63
Figure 35 – Cyclic Read Data Block		65
Figure 36 – Cyclic Read Data Block example		65
Figure 37 – Event Data Block.....		66
Figure 38 – Service Data Block.....		68

Figure 39 – Fixed Controller-to-Device Connection format (fixed size = 16 bytes)	69
Figure 40 – Fixed Device-to-Controller Connection format (fixed size = 16 bytes)	69
Figure 41 – CIP Motion 1-Cycle timing model	70
Figure 42 – CIP Motion 2-Cycle timing model	72
Figure 43 – CIP Motion 3-Cycle timing model	73
Figure 44 – Controller-to-Device Connection timing with fine interpolation	74
Figure 45 – Controller-to-Device Connection timing with extrapolation	76
Figure 46 – Use of Time Stamp to adjust actual position to the controller's timebase	77
Figure 47 – Coordination of two drives with different Update Periods	79
Figure 48 – Coordination of multiple drive axes in case of delayed Controller-to-Device Connection packets	80
Figure 49 – Propagation of a step change in time	81
Figure 50 – Configuration Block Format Revision 1 (Connection Point 81)	86
Figure 51 – Configuration Block Format Revision 2 (Connection Point 82)	87
Figure 52 – Typical initial C-to-D connection data block	88
Figure 53 – Typical initial D-to-C connection data block	88
Figure 54 – Typical contents of first C-to-D class attribute configuration packet	88
Figure 55 – Typical response to first C-to-D class configuration packet	89
Figure 56 – Typical contents of first C-to-D axis instance configuration packet	89
Figure 57 – Typical response to first C-to-D axis configuration packet	90
Figure 58 – Typical contents of C-to-D Time Sync service request packet	90
Figure 59 – Group Sync of CIP Motion devices	91
Figure 60 – Object components for CIP Motion control architecture	94
Figure 61 – Command Control Word field	127
Figure 62 – IEEE Std 112 per phase motor model	130
Figure 63 – Event Checking Control Word field	152
Figure 64 – Event Checking Status word field	153
Figure 65 – Brake Control Sequence (Category 0 Stop)	182
Figure 66 – Brake Control Sequence (Category 1 Stop)	183
Figure 67 – Brake Control Sequence (Category 2 Stop)	184
Figure 68 – Drive Enable sequence with Proving feature	185
Figure 69 – Drive Disable sequence with Proving feature	186
Figure 70 – Get_Axis_Attributes_List Request format	227
Figure 71 – Get_Axis_Attributes_List Response format	228
Figure 72 – Get_Axis_Attributes_List Response – Single 4-byte attribute	228
Figure 73 – Get_Axis_Attributes_List Response – Single 2-byte attribute	228
Figure 74 – Get_Axis_Attributes_List Response – Byte attribute array	229
Figure 75 – Get_Axis_Attributes_List Response – Two Dimensional attribute array	229
Figure 76 – Get_Axis_Attributes_List Response – Error example	229
Figure 77 – Set_Axis_Attributes_List Request format	230
Figure 78 – Set_Axis_Attributes_List Request – Single 4-byte attribute	230
Figure 79 – Set_Axis_Attributes_List Request – Single 2-byte attribute	231
Figure 80 – Set_Axis_Attributes_List Request – 2-byte attribute array	231

Figure 81 – Set_Axis_Attributes_List Request – Two dimensional attribute array	231
Figure 82 – Set_Axis_Attributes_List Response format	231
Figure 83 – Set_Cyclic_Write_List Request format	232
Figure 84 – Set_Cyclic_Write_List Response format	232
Figure 85 – Set_Cyclic_Read_List Request format	233
Figure 86 – Set_Cyclic_Read_List Response format	233
Figure 87 – Motion Device Axis Object State Model	241
Figure 88 – Motion Device Axis Object State Model for Feedback Only	243
Figure 89 – Motion Device Axis Object State Model for Converter	244
Figure 90 – Command Generator	267
Figure 91 – Feedback Channels 1 and 2	272
Figure 92 – Event Capture Functionality	273
Figure 93 – No Control (Feedback Only)	275
Figure 94 – Closed Loop Position Control	276
Figure 95 – Closed Loop Velocity Control	278
Figure 96 – Open Loop Frequency Control	280
Figure 97 – Acceleration Control	282
Figure 98 – Torque Control	282
Figure 99 – Closed Loop Current Vector Control	286
Table 1 – Data types	32
Table 2 – Objects present in a CIP Motion device	33
Table 3 – Motion Device Axis Object content by Device Type	34
Table 4 – Object effect on behavior	34
Table 5 – Object interfaces	35
Table 6 – Time Data Set	41
Table 7 – Axis Control	45
Table 8 – Control Status	45
Table 9 – Command Data Set	46
Table 10 – Command Data Element to Motion Device Axis Object attribute mapping	46
Table 11 – Actual Data Set	47
Table 12 – Actual Data Element to Motion Device Axis Object attribute Mapping	47
Table 13 – Status Data Set	48
Table 14 – Command Control	48
Table 15 – Command Target Update vs. Update Period Ratio	49
Table 16 – Basic Event Cycle	51
Table 17 – Extended Event Cycle	53
Table 18 – Basic Event Cycle with Auto-rearm	55
Table 19 – Registration Data Set	57
Table 20 – Home Data Set	58
Table 21 – Watch Data Set	58
Table 22 – Axis Response	64
Table 23 – Event Type	67

Table 24 – Propagation of a step change in time (example 1)	81
Table 25 – Propagation of a step change in time (example 2)	83
Table 26 – CIP Motion visualisation components	92
Table 27 – Motion Device Axis Object revision history	93
Table 28 – Example for instance attribute implementation vs. Device Function Code	96
Table 29 – Instance attribute implementation vs. Device Function Code	98
Table 30 – Class attributes for the Motion Device Axis Object.....	108
Table 31 – Node Control bit definitions	111
Table 32 – Node Status bit definitions.....	112
Table 33 – Node Fault Code definitions	113
Table 34 – Node Alarm Code definitions	114
Table 35 – Dynamic Units vs. Feedback Mode	116
Table 36 – Motion Control configuration attributes	116
Table 37 – Control Mode enumeration definitions.....	117
Table 38 – Control Method enumeration definitions.....	117
Table 39 – Motion Scaling attributes	118
Table 40 – Motion Unit selection rules	120
Table 41 – Signal attributes affected by Motion Polarity	121
Table 42 – Directional Limit attributes affected by Motion Polarity.....	123
Table 43 – Connection Data attributes	124
Table 44 – Actual Data Set value determination.....	126
Table 45 – Command Data Set value determination.....	127
Table 46 – Command Target Update enumeration definition	127
Table 47 – Command Position Data Type enumeration definition	128
Table 48 – Status Data Set bit definitions	128
Table 49 – Registration Event Data format.....	129
Table 50 – Home Event Data format	129
Table 51 – Watch Event Data format.....	129
Table 52 – General Motor Info attributes	130
Table 53 – General Motor Configuration attributes	131
Table 54 – General PM Motor Configuration attributes	134
Table 55 – General Rotary Motor Configuration attributes	135
Table 56 – General Linear Motor Configuration attributes	136
Table 57 – Rotary PM Motor Configuration attributes	137
Table 58 – Linear PM Motor Configuration attributes.....	137
Table 59 – Induction Motor Configuration attributes	138
Table 60 – Load Transmission and Actuator Configuration attributes	139
Table 61 – Feedback Types abbreviations	140
Table 62 – Logical Feedback Channel Control functions	140
Table 63 – Logical Feedback Channel rules.....	141
Table 64 – General Feedback Info attributes.....	142
Table 65 – General Feedback Signal attributes.....	142
Table 66 – Feedback Configuration attributes	143

Table 67 – Feedback Mode enumeration definitions.....	149
Table 68 – Event attributes	150
Table 69 – Event Checking Control bit definitions	152
Table 70 – Event Checking Status bit definitions.....	154
Table 71 – Command Generator Signal attributes	155
Table 72 – Command Generator Configuration attributes	157
Table 73 – Position Loop Signal attributes	159
Table 74 – Position Loop Configuration attributes	160
Table 75 – Velocity Loop Signal attributes	162
Table 76 – Velocity Loop Configuration attributes	163
Table 77 – Acceleration Signal attributes	165
Table 78 – Acceleration Configuration attributes	165
Table 79 – Torque/Force Control Signal attributes	166
Table 80 – Torque/Force Control Configuration attributes	167
Table 81 – Current Control Signal attributes	169
Table 82 – Current Control Configuration attributes	171
Table 83 – Frequency Control Signal attributes.....	175
Table 84 – Frequency Control Configuration attributes.....	175
Table 85 – Drive Output attributes	176
Table 86 – Stopping/Braking attributes	177
Table 87 – Stopping Action enumeration definitions.....	180
Table 88 – Proving sub-feature attribute dependencies.....	184
Table 89 – DC Bus Control attributes	187
Table 90 – Power and Thermal Management Status attributes	190
Table 91 – Power and Thermal Management Configuration attributes	192
Table 92 – Axis Status attributes	194
Table 93 – Axis Status bit definitions	195
Table 94 – Axis Status bit vs. Axis State	198
Table 95 – Stopping Action vs. Stop Category	199
Table 96 – Axis I/O Status bit definitions.....	199
Table 97 – Exception, Fault and Alarm attributes	200
Table 98 – Standard Exception Table	202
Table 99 – Fault and Alarm Log attributes.....	207
Table 100 – Exception Factory Limit Info attributes	210
Table 101 – Exception User Limit Configuration attributes	211
Table 102 – Axis Exception Action Configuration attribute	213
Table 103 – Axis Exception Action definitions	214
Table 104 – Initialization Fault attributes.....	216
Table 105 – Standard Initialization Fault Table	216
Table 106 – Start Inhibit attributes	217
Table 107 – Standard Start Inhibit Table	217
Table 108 – APR Fault attributes	218
Table 109 – Standard APR Fault Table	219

Table 110 – Axis Statistical attributes	219
Table 111 – Axis Info attributes.....	220
Table 112 – Drive General Purpose I/O attributes	221
Table 113 – Local Mode Configuration attributes	222
Table 114 – Axis Safety Status attributes.....	223
Table 115 – Motion Device Axis Object – Common Services	224
Table 116 – Group_Sync Request Data Structure	224
Table 117 – Group_Sync Response Data Structure	225
Table 118 – Motion Device Axis Object – Object Specific Services	226
Table 119 – Run_Motor_Test Request structure	234
Table 120 – Get_Motor_Test_Data measured by Test Type	235
Table 121 – Get_Motor_Test_Data Request structure (optional)	235
Table 122 – Get_Motor_Test_Data Response standard structure (Motor Type = Induction)	236
Table 123 – Get_Motor_Test_Data Response standard structure (Motor Type = SPM)	236
Table 124 – Get_Motor_Test_Data Response standard structure (Motor Type = IPM).....	237
Table 125 – Run_Inertia_Test Request structure	237
Table 126 – Get_Inertia_Test_Data Response structure	238
Table 127 – Run_Hookup_Test Request structure	239
Table 128 – Get_Hookup_Test_Data measured by Test Type	240
Table 129 – Get_Hookup_Test_Data Response structure	240
Table 130 – Axis State Machine transitions.....	242
Table 131 – Axis State Machine conditions	243
Table 132 – Axis State Machine transitions (Feedback Only)	244
Table 133 – Axis State Machine transitions (Converter)	245
Table 134 – Axis Control Request code	246
Table 135 – Axis Response Acknowledge codes.....	246
Table 136 – Completion criteria for requested operation	247
Table 137 – Possible error conditions for requested operation	247
Table 138 – Successful Axis Control Request Cycle	248
Table 139 – Unsuccessful Axis Control Request Cycle	248
Table 140 – Pending Axis Control Request Cycle	249
Table 141 – Cancel Request Cycle	250
Table 142 – Redefine Position Reference Cycle.....	252
Table 143 – Running State – Configurable attributes	255
Table 144 – Axis state mapping to Identity Object with LED behavior	262
Table 145 – CIP Motion Device seven-segment display behavior	263
Table 146 – CIP Motion multi-character alphanumeric display behavior	264
Table 147 – Multi-axis multi-character alphanumeric display behavior	266

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ADJUSTABLE SPEED ELECTRICAL
POWER DRIVE SYSTEMS –****Part 7-202: Generic interface and use of profiles for
power drive systems – Profile type 2 specification**

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 61800-7-202 has been prepared by subcommittee SC 22G: Adjustable speed electric drive systems incorporating semiconductor power converters, of IEC technical committee TC 22: Power electronic systems and equipment.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision.

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

- a) update of patent information;
- b) new revision of the Drive Profile and Drive Axis specifications, with multiple clarifications and enhancements.

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

FDIS	Report on voting
22G/308/FDIS	22G/323/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 parts of the IEC 61800 series, under the general title *Adjustable speed electrical power drive systems*, can be found on the IEC website.

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

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

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

INTRODUCTION

0.1 General

The IEC 61800 series is intended to provide a common set of specifications for adjustable speed electrical power drive systems.

IEC 61800-7 specifies profiles for Power Drive Systems (PDS) and their mapping to existing communication systems by use of a generic interface model.

IEC 61800-7 describes a generic interface between control systems and power drive systems. This interface can be embedded in the control system. The control system itself can also be located in the drive (sometimes known as "smart drive" or "intelligent drive").

A variety of physical interfaces is available (analogue and digital inputs and outputs, serial and parallel interfaces, fieldbuses and networks). Profiles based on specific physical interfaces are already defined for some application areas (e.g. motion control) and some device classes (e.g. standard drives, positioner). The implementations of the associated drivers and application programmers interfaces are proprietary and vary widely.

IEC 61800-7 defines a set of common drive control functions, parameters, and state machines or description of sequences of operation to be mapped to the drive profiles.

IEC 61800-7 provides a way to access functions and data of a drive that is independent of the used drive profile and communication interface. The objective is a common drive model with generic functions and objects suitable to be mapped on different communication interfaces. This makes it possible to provide common implementations of motion control (or velocity control or drive control applications) in controllers without any specific knowledge of the drive implementation.

There are several reasons to define a generic interface:

For a drive device manufacturer

- less effort to support system integrators;
- less effort to describe drive functions because of common terminology;
- the selection of drives does not depend on availability of specific support.

For a control device manufacturer

- no influence of bus technology;
- easy device integration;
- independent of a drive supplier.

For a system integrator

- less integration effort for devices;
- only one understandable way of modeling;
- independent of bus technology.

Much effort is needed to design a motion control application with several different drives and a specific control system. The tasks to implement the system software and to understand the functional description of the individual components may exhaust the project resources. In some cases, the drives do not share the same physical interface. Some control devices just support a single interface which will not be supported by a specific drive. On the other hand, the functions and data structures are often specified with incompatibilities. This requires the

system integrator to write special interfaces for the application software and this should not be his responsibility.

Some applications need device exchangeability or integration of new devices in an existing configuration. They are faced with different incompatible solutions. The efforts to adapt a solution to a drive profile and to manufacturer specific extensions may be unacceptable. This will reduce the degree of freedom to select a device best suited for this application to the selection of the unit which will be available for a specific physical interface and supported by the controller.

IEC 61800-7-1 is divided into a generic part and several annexes as shown in Figure 1. The drive profiles types for CiA® 402¹, CIP Motion™², PROFIdrive³ and SERCOS®⁴ are mapped to the generic interface in the corresponding annex. The annexes have been submitted by open international network or fieldbus organizations which are responsible for the content of the related annex and use of the related trademarks.

This part of IEC 61800-7 specifies the profile type 2 (CIP Motion™).

The profile types 1, 3 and 4 are specified in IEC 61800-7-201, IEC 61800-7-203 and IEC 61800-7-204.

¹ CiA® 402 is a registered trade mark of CAN in Automation e.V (CiA). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the registered trade mark CiA® 402. Use of the registered trade mark CiA® 402 requires permission of CAN in Automation e.V (CiA).

² CIP Motion™ is a trade mark of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark CIP Motion™. Use of the trade mark CIP Motion™ requires permission of ODVA, Inc.

³ PROFIdrive is a trade name of PROFIBUS & PROFINET International. 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 profile does not require use of the trade name PROFIdrive. Use of the trade name PROFIdrive requires permission of PROFIBUS & PROFINET International.

⁴ SERCOS® is a registered trade mark of SERCOS International e.V. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the registered trade mark SERCOS®. Use of the registered trade mark SERCOS® requires permission of the trade mark holder.

IEC 61800-7-301, IEC 61800-7-302, IEC 61800-7-303 and IEC 61800-7-304 specify how the profile types 1, 2, 3 and 4 are mapped to different network technologies (such as CANopen®⁵, CC-Link IE® Field Network⁶, EPA™⁷, EtherCAT®⁸, Ethernet Powerlink™⁹, DeviceNet™¹⁰, ControlNet™¹¹, EtherNet/IP™¹², PROFIBUS¹³, PROFINET¹⁴ and SERCOS®).

-
- ⁵ CANopen® is a registered trade mark of CAN in Automation e.V. (CiA). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the registered trade mark CANopen®. Use of the registered trade mark CANopen® requires permission of CAN in Automation e.V. (CiA). CANopen® is an acronym for Controller Area Network *open* and is used to refer to EN 50325-4.
- ⁶ CC-Link IE® Field Network is a registered trade mark of Mitsubishi Electric Corporation. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the registered trade mark CC-Link IE® Field Network. Use of the registered trade mark CC-Link IE® Field Network requires permission of Mitsubishi Electric Corporation.
- ⁷ EPA™ is a trade mark of SUPCON Group Co. Ltd. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark EPA™. Use of the trade mark EPA™ requires permission of the trade mark holder.
- ⁸ EtherCAT® is a registered trade mark of Beckhoff, Verl. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the registered trade mark EtherCAT®. Use of the registered trade mark EtherCAT® requires permission of the trade mark holder.
- ⁹ Ethernet Powerlink™ is a trade mark of Bernecker & Rainer Industrieelektronik Ges.m.b.H., control of trade mark use is given to the non profit organization EPSG. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark Ethernet Powerlink™. Use of the trade mark Ethernet Powerlink™ requires permission of the trade mark holder.
- ¹⁰ DeviceNet™ is a trade mark of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark DeviceNet™. Use of the trade mark DeviceNet™ requires permission of ODVA, Inc.
- ¹¹ ControlNet™ is a trade mark of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark ControlNet™. Use of the trade mark ControlNet™ requires permission of ODVA, Inc.
- ¹² EtherNet/IP™ is a trade mark of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark EtherNet/IP™. Use of the trade mark EtherNet/IP™ requires permission of ODVA, Inc.
- ¹³ PROFIBUS is a trade name of PROFIBUS & PROFINET International. 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 profile does not require use of the trade name PROFIBUS. Use of the trade name PROFIBUS requires permission of PROFIBUS & PROFINET International.
- ¹⁴ PROFINET is a trade name of PROFIBUS & PROFINET International. 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 profile does not require use of the trade name PROFIBUS & PROFINET. Use of the trade name PROFIBUS & PROFINET requires permission of PROFIBUS & PROFINET International.

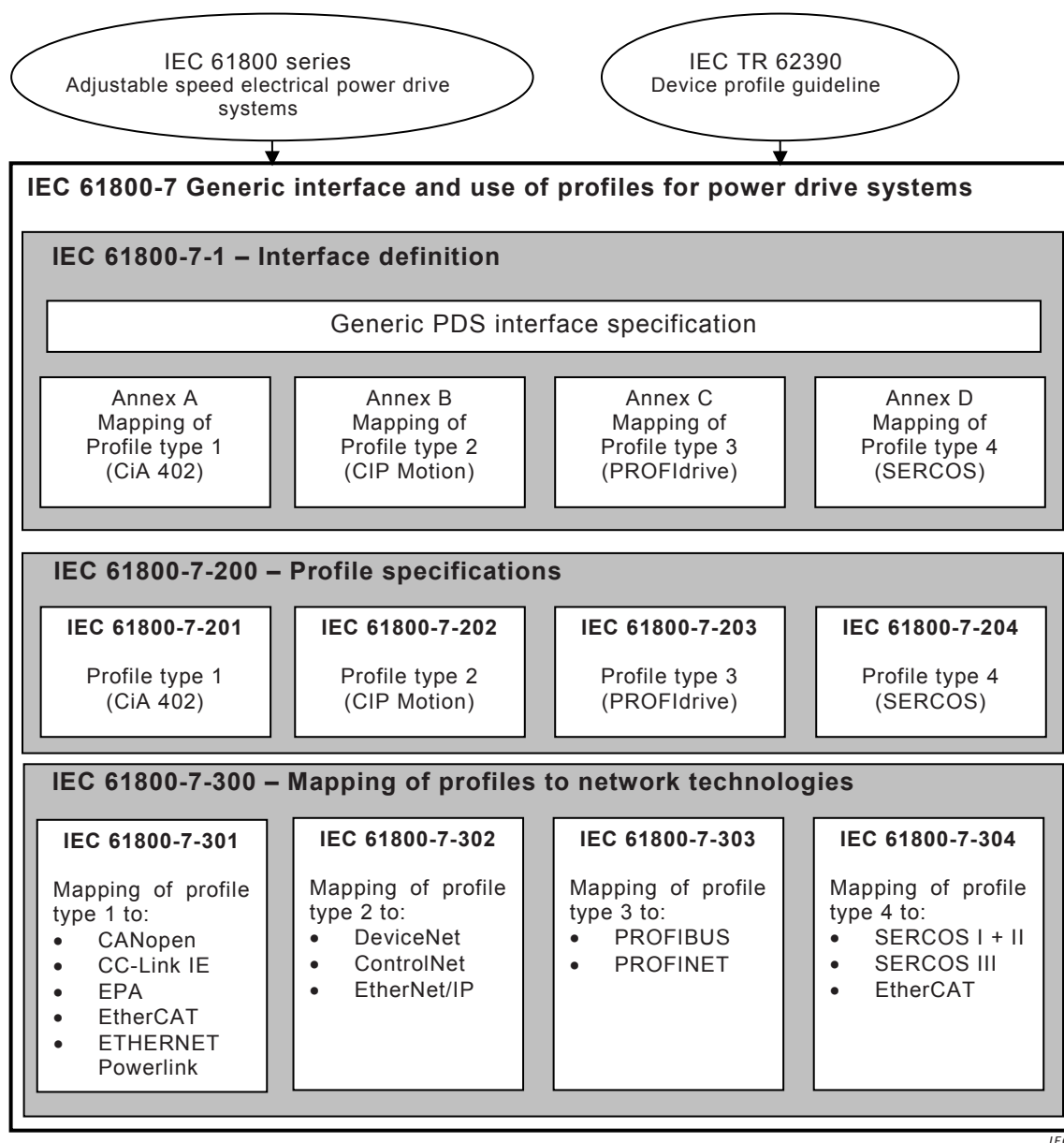


Figure 1 – Structure of IEC 61800-7

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 a patent concerning the following. This patent is held by its inventors under license to ODVA, Inc:

Publication / Application serial number	Holder	Title
US 7,983,769 EP 1659465	[ODVA]	Time stamped motion control network protocol that enables balanced single cycle timing and utilization of dynamic data structures

IEC takes no position concerning the evidence, validity and scope of this patent right.

ODVA and the holder of this patent right have assured the IEC that ODVA is 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 statement of ODVA and the holder of this patent right is registered with IEC. Information may be obtained from:

[ODVA]	ODVA, Inc. 2370 East Stadium Boulevard #1000 Ann Arbor, Michigan 48104 USA Attention: Office of the Executive Director email: odva@odva.org
--------	---

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 data bases for the most up to date information concerning patents.

ADJUSTABLE SPEED ELECTRICAL POWER DRIVE SYSTEMS –

Part 7-202: Generic interface and use of profiles for power drive systems – Profile type 2 specification

1 Scope

This part of IEC 61800 specifies profile type 2 (CIP Motion™) for Power Drive Systems (PDS). Profile type 2 can be mapped onto different communication network technologies.

The functions specified in this part of IEC 61800-7 are not intended to ensure functional safety. This requires additional measures according to the relevant standards, agreements and laws.

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 60204-1, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

IEC 61158-4-2:2014, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification – Type 2 elements*

IEC 61158-5-2:2014, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61158-6-2:2014, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification – Type 2 elements*

IEC 61588:2009, *Precision clock synchronization protocol for networked measurement and control systems*

IEC 61800-7-1:2015, *Adjustable speed electrical power drive systems – Part 7-1: Generic interface and use of profiles for power drive systems – Interface definition*

IEEE Std 112-2004, *IEEE Standard Test Procedure for Polyphase Induction Motors and Generators*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1**algorithm**

completely determined finite sequence of operations by which the values of the output data can be calculated from the values of the input data

[SOURCE: IEC 61800-7-1:2015, 3.2.1]

3.1.2**application**

software functional element specific to the solution of a problem in industrial-process measurement and control

Note 1 to entry: An application may be distributed among resources, and may communicate with other applications.

[SOURCE: IEC 61800-7-1:2015, 3.2.2]

3.1.3**application mode**

type of application that can be requested from a PDS

Note 1 to entry: The different application modes reflect the control loop for torque control, velocity control, position control or other applications such as homing.

[SOURCE: IEC 61800-7-1:2015, 3.3.1.2]

3.1.4**attribute**

property or characteristic of an entity

[SOURCE: IEC 61800-7-1:2015, 3.2.3]

3.1.5**axis**

logical element inside an automation system (e.g. a motion control system) that represents some form of movement

Note 1 to entry: Axes can be rotary or linear, physical or virtual, controlled or simply observed.

[SOURCE: IEC 61800-7-1:2015, 3.2.4]

3.1.6**bus regulator**

any method used to limit the rise in DC bus voltage level that occurs when decelerating a motor

3.1.7**CIP Motion™¹⁵**

extensions to the CIP services and protocol to support motion control over CIP networks

[SOURCE: IEC 61800-7-1:2015, 3.3.3.1]

¹⁵ CIP Motion™ and CIP Sync™ are trade marks of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade marks CIP Motion™ or CIP Sync™. Use of the trade marks CIP Motion™ or CIP Sync™ requires permission of ODVA, Inc.

3.1.8**CIP Motion™ controller**

CIP compliant controller containing a Motion Control Axis Object that can interface to a CIP Motion device via a CIP Motion I/O Connection

Note 1 to entry: A description of the Motion Control Axis Object is beyond the scope of IEC 61800-7.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.2]

3.1.9**CIP Motion™ device**

CIP compliant device containing one or more Motion Device Axis Object instances that can communicate to a CIP Motion controller via a CIP Motion I/O Connection

EXAMPLE: A CIP Motion drive is a particular case of a CIP Motion device.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.3]

3.1.10**CIP Motion™ drive profile**

collection of objects used to implement a CIP Motion drive device that includes the Motion Device Axis Object, as well as standard support objects like the Identity Object and the Time Sync Object

Note 1 to entry: The Device Type assigned to the CIP Motion drive profile is 25_{hex}.

3.1.11**CIP Motion™ I/O Connection****CIP Motion™ Connection**

periodic bi-directional, class 1, CIP connection between a controller and a drive that is defined as part of the CIP Motion specification

[SOURCE: IEC 61800-7-1:2015, 3.3.3.4]

3.1.12**CIP Sync™¹⁵**

extensions to the CIP services and protocol to encapsulate IEC 61588:2009 time synchronization functionality over a CIP Network

Note 1 to entry: See Time Sync Object in IEC 61158-5-2 and IEC 61158-6-2.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.5]

3.1.13**class**

description of a set of objects that share the same attributes, operations, methods, relationships, and semantics

[SOURCE: IEC 61800-7-1:2015, 3.2.5]

3.1.14**closed loop**

methods of control where there is a feedback signal of some kind that is used to drive the actual dynamics of the motor to match the commanded dynamics by servo action

Note 1 to entry: In most cases, there is a literal feedback device to provide this signal, but in some cases, the signal is derived from the motor excitation (i.e. sensorless operation).

3.1.15**commands**

set of commands from the application control program to the PDS to control the behavior of the PDS or functional elements of the PDS

Note 1 to entry: The behavior is reflected by states or operating modes.

Note 2 to entry: The different commands may be represented by one bit each.

[SOURCE: IEC 61800-7-1:2015, 3.3.1.3]

3.1.16**control**

purposeful action on or in a process to meet specified objectives

[SOURCE: IEC 61800-7-1:2015, 3.2.6]

3.1.17**control device**

physical unit that contains – in a module/subassembly or device – an application program to control the PDS

[SOURCE: IEC 61800-7-1:2015, 3.2.7]

3.1.18**converter**

device that generally converts AC input to DC output

Note 1 to entry: A converter is also commonly called the Drive Power Supply. In the context of a drive system, the converter is responsible for converting AC main input into DC bus power.

3.1.19**Cyclic Data Block**

high priority real-time data block that is transferred by a CIP Motion Connection on a periodic basis

3.1.20**data type**

set of values together with a set of permitted operations

[SOURCE: IEC 61800-7-1:2015, 3.2.8]

3.1.21**device**

field device

<function blocks> networked independent physical entity of an industrial automation system capable of performing specified functions in a particular context and delimited by its interfaces

[SOURCE: IEC 61800-7-1:2015, 3.2.9]

3.1.22**device**

field device

<system integration> entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system

[SOURCE: IEC 61800-7-1:2015, 3.2.10]

3.1.23**device profile**

representation of a device in terms of its parameters, parameter assemblies and behaviour according to a device model that describes the data and behaviour of the device as viewed through a network, independent from any network technology

[SOURCE: IEC 61800-7-1:2015, 3.2.11]

3.1.24**drive**

device designed to control the dynamics of a motor

3.1.25**Event Data Block**

medium priority real-time data block that is transferred by a CIP Motion Connection only after a specified event occurs

Note 1 to entry: Registration and marker input transitions are typical drive events.

3.1.26**functional element**

entity of software or software combined with hardware, capable of accomplishing a specified function of a device

Note 1 to entry: A functional element has an interface, associations to other functional elements and functions.

Note 2 to entry: A functional element can be made out of function block(s), object(s) or parameter list(s).

[SOURCE: IEC 61800-7-1:2015, 3.2.13]

3.1.27**input data**

data transferred from an external source into a device, resource or functional element

[SOURCE: IEC 61800-7-1:2015, 3.2.14]

3.1.28**interface**

shared boundary between two entities defined by functional characteristics, signal characteristics, or other characteristics as appropriate

[SOURCE: IEC 61800-7-1:2015, 3.2.15]

3.1.29**inverter**

device that generally converts DC input to AC output

Note 1 to entry: An inverter is also commonly called the Drive Amplifier. In the context of a drive system, the inverter is responsible for controlling the application of DC bus power to an AC motor.

3.1.30**model**

mathematical or physical representation of a system or a process, based with sufficient precision upon known laws, identification or specified suppositions

[SOURCE: IEC 61800-7-1:2015, 3.2.17]

3.1.31**motion**

any aspect of the dynamics of an axis

Note 1 to entry: In the context of this part of IEC 61800-7, it is not limited to servo drives but encompasses all forms of drive based motor control.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.8]

3.1.32**Motion Control Axis Object**

object that defines the attributes, services, and behavior of a controller based axis according to the CIP Motion specification

3.1.33**Motion Device Axis Object**

object that defines the attributes, services, and behavior of a motion device based axis according to the CIP Motion specification

Note 1 to entry: This object includes Communications, Device control, and Basic drive FE elements as defined in IEC 61800-7.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.9]

3.1.34**open loop**

methods of control where there is no application of feedback to force the actual motor dynamics to match the commanded dynamics

EXAMPLE Examples of open loop control are stepper drives and variable frequency drives.

3.1.35**operating mode**

characterization of the way and the extent to which the human operator intervenes in the control equipment

[SOURCE: IEC 61800-7-1:2015, 3.2.18]

3.1.36**output data**

data originating in a device, resource or functional element and transferred from them to external systems

[SOURCE: IEC 61800-7-1:2015, 3.2.19]

3.1.37**parameter**

data element that represents device information that can be read from or written to a device, for example through the network or a local HMI

Note 1 to entry: A parameter is typically characterized by a parameter name, data type and access direction.

[SOURCE: IEC 61800-7-1:2015, 3.2.20]

3.1.38**profile**

representation of a PDS interface in terms of its parameters, parameter assemblies and behavior according to a communication profile and a device profile

[SOURCE: IEC 61800-7-1:2015, 3.2.21, modified – Note 1 to entry is deleted]

3.1.39

read

get

operation that involves the retrieving of an attribute value from the perspective of the controller side of the interface

3.1.40

registration

high speed record of motion axis position triggered by an event

3.1.41

Service Data Block

lower priority real-time data block associated with a service message from the controller that is transferred by a CIP Motion Connection on a periodic basis

Note 1 to entry: Service data includes service request messages to access Motion Device Axis Object attributes or perform various drive diagnostics.

3.1.42

set-point

value or variable used as output data of the application control program to control the PDS

Note 1 to entry: The value sent to the drive is used to directly control some aspect of the motor dynamics, which includes (but is not limited to) position, velocity, acceleration, and torque.

[SOURCE: IEC 61800-7-1:2015, 3.3.1.5, modified – Note 1 to entry is added]

3.1.43

shunt regulator

specific bus regulator method that switches the DC bus across a power dissipating resistor to dissipate the regenerative power of a decelerating motor

3.1.44

status

set of information from the PDS to the application control program reflecting the state or mode of the PDS or a functional element of the PDS

Note 1 to entry: The different status information may be coded with one bit each.

[SOURCE: IEC 61800-7-1:2015, 3.3.1.6]

3.1.45

synchronized

condition where the local clock value on the drive is locked onto the master clock of the distributed System Time

Note 1 to entry: When synchronized, the drive and controller devices may utilise time stamps associated with CIP Motion Connection data.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.11]

3.1.46

System Time

absolute time value as defined in the CIP Sync specification in the context of a distributed time system where all devices have a local clock that is synchronized with a common master clock

Note 1 to entry: In the context of CIP Motion, System Time is a 64-bit integer value in units of nanoseconds with a value of 0 corresponding to the date 1970-01-01.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.12]

3.1.47

thermal model

any algorithm that attempts to model the thermal behavior of the associated device during operation

3.1.48

time offset

System Time offset value associated with the CIP Motion Connection data that is associated with source device

Note 1 to entry: The System Time offset is a 64-bit offset value that is added to a device's local clock to generate System Time for that device

3.1.49

time stamp

System Time stamp value associated with the CIP Motion Connection data that conveys the absolute time when the associated data was captured, or that can also be used to determine when the associated data shall be applied

Note 1 to entry: CIP time stamps are always in the context of a distributed time system where all nodes on the CIP control network have clocks that are synchronized with a master clock source using CIP Sync.

[SOURCE: IEC 61800-7-1:2015, 3.3.3.13]

3.1.50

type

hardware or software element which specifies the common attributes shared by all instances of the type

[SOURCE: IEC 61800-7-1:2015, 3.2.23]

3.1.51

variable

software entity that may take different values, one at a time

Note 1 to entry: The values of a variable as well as of a parameter are usually restricted to a certain data type.

[SOURCE: IEC 61800-7-1:2015, 3.2.25]

3.1.52

variable frequency drive

VFD

class of drive products that seek to control the speed of a motor, typically an induction motor, through a proportional relationship between drive output voltage and commanded output frequency

Note 1 to entry: Variable frequency drives are therefore sometimes referred to as a Volts/Hertz drives.

Note 2 to entry: The English abbreviation VFD is also used in French.

3.1.53

vector drive

class of drive products that seek to control the dynamics of a motor via closed loop control which includes, but is not limited to, closed loop control of both torque and flux vector components of motor stator current relative to the rotor flux vector

3.1.54**write**

set

operation that involves the setting of an attribute to a specified value from the perspective of the controller side of the interface

3.2 Abbreviated terms

Ack	acknowledge
Act	actual
Blk	block
CIP™ ¹⁶	Common Industrial Protocol (see IEC 61158 Type 2, IEC 61784-1 and IEC 61784-2 Communication Profile Family 2)
Cmd	command
Cyc	cyclic
HMI	human machine interface
ID	identifier
impl.	implementation
I/O	Input/Output
Opt.	optional
PDS	power drive system
Req.	required
RMS	root mean square
r/min	revolutions per minute
Sync'd	synchronized
TCP	Transmission Control Protocol (see IETF RFC 793)
UDP	User Datagram Protocol (see IETF RFC 768)
VFD	variable frequency drive

4 Overview**4.1 General**

CIP Motion devices control, monitor or support the motion of one or more moving component of a machine. Machine motion is typically generated by rotary or linear motion actuators, i.e.: motors, and monitored by feedback devices. Each motor is typically driven by a power structure and a motion control algorithm that comprise a CIP Motion Drive Device Type.

Drive functionality supported by the CIP Motion drive device profile can be applied to a variety of motor technologies, and can range from very simple “open loop” variable frequency drives to sophisticated “closed loop” vector controlled servo drives. In either case, motion is controlled via a command reference that can be configured for position control, velocity control, acceleration control, or current/torque control. The CIP Motion drive device profile also supports position, velocity and acceleration monitoring through multiple feedback, as does the CIP Motion encoder device profile.

¹⁶ CIP™ is a trade mark of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade mark holder or any of its products. Compliance to this profile does not require use of the trade mark CIP™. Use of the trade mark CIP™ requires permission of ODVA, Inc.

All the attributes, services, and state behavior of a CIP Motion device are encapsulated in one or more Motion Device Axis Object instances. In addition to motion control and feedback monitoring functionality, the Motion Device Axis Object includes support for event monitoring, such position capture on a Registration event, and DC Bus management associated with a Power Converter. A CIP Motion device can manifest any combination of these functions to create various classes of CIP Motion compliant devices ranging from full featured servo drives, to CIP Motion Encoders, to standalone Power Converters, differentiated by the CIP Motion Device Type.

This CIP Motion drive profile specification and the associated Motion Device Axis Object specification define the interface, the specific attributes, and command behaviors required to support motion control when connected to a CIP Motion compliant controller through a CIP network.

Of the applicable CIP Networks, EtherNet/IP™ is the network of choice for high performance, synchronized multi-axis control. Other CIP networks such as ControlNet™ and DeviceNet™ could be applied to lower performance non-synchronized motion device applications such as simple variable frequency drives, velocity loop drives, and indexing drives.

4.2 Control modes

4.2.1 General

The Motion Device Axis Object covers the behavior of various motion control system devices that includes feedback devices, drive devices, standalone converters and motion I/O devices. For drive devices, the Motion Device Axis Object covers a wide range of drive types from simple variable frequency (V/Hz) drives, to sophisticated position control servo drives, with or without integral converters. Indeed, many commercial drive products can be configured to operate in any one of these different motion control modes depending on the specific application requirements. The attributes of the Motion Device Axis Object are therefore organized to address this broad range of functionality and the framework for this organization is described in 4.2.

IEC 61800-7-1 and this part of the IEC 61800-7 series are organised around the general philosophy that position control is the highest form of dynamic control. That is, position control implies velocity control, and velocity control implies acceleration control. Acceleration is related to torque or force by the inertia or mass of the load, respectively, acceleration control implies torque control. And finally, since motor torque or force is generally related to motor current by a torque or force constant, respectively, torque control implies current control. The torque or force constant can be a function of the motor magnets as in a Permanent Magnet motor, or the induced flux of an Induction motor.

Since acceleration, torque/force, and current are generally related by a constant, these terms are sometimes used interchangeably in the industry. For example IEC 61800-7-1 refers to a torque control loop rather than a current control loop. This specification attempts to differentiate between these control properties where applicable. This is particularly useful when the relationship between them is not static, such as when inertia/mass changes with position or time, or when the torque/force constant changes due to temperature change or motor flux variation.

4.2.2 Control methods

Within this basic control paradigm, there is latitude for different control methods, both closed loop and open loop. By closed loop, it is generally implied that there is a feedback signal that is used to drive the actual dynamics of the motor to match the commanded dynamics by servo action. In most cases, there is a literal feedback device to provide this signal, and in some cases, the signal is derived from the motor excitation (i.e. sensorless/encoderless operation). By open loop, it is implied that there is no application of feedback to force the actual dynamics to match the commanded dynamics. While precision and performance are the hallmarks of closed loop control, simplicity and economy are the hallmarks of open loop control.

4.2.3 Control nomenclature

Finally, as evident in the description above, linear and rotary control applications can affect the control nomenclature. While rotary applications speak of torque and inertia, linear applications speak of force and mass. For the purposes of this part of the IEC 61800-7 series, when referring to rotary nomenclature, the defined behavior can generally be applied to linear applications by substituting the terms, force for torque and mass for inertia. With that understanding, Figure 2 to Figure 8 follow the IEC 61800-7-1 nomenclature using torque rather than force without loss of generality.

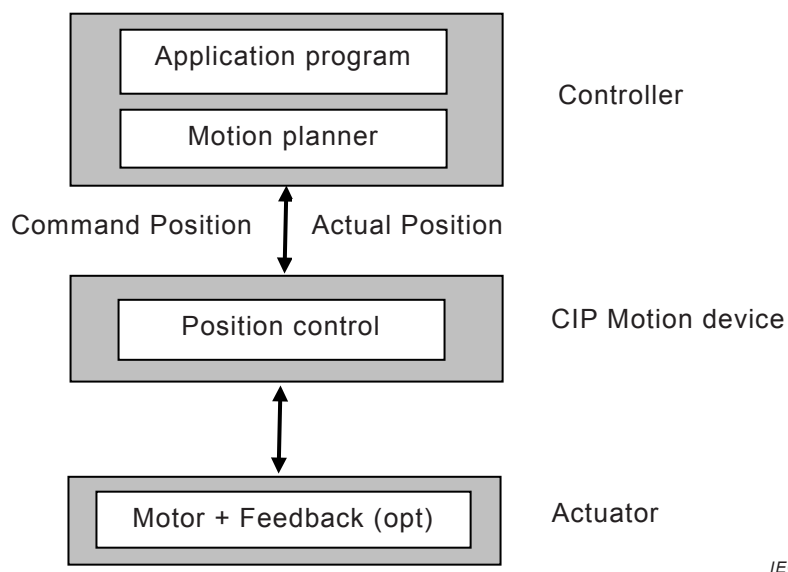
4.2.4 Position control

4.2.4.1 General concepts

In position control application mode, either the application control program (command execution function) or the motion planner (move trajectory control function) provide a set-point value to the CIP Motion device via the cyclic data connection. The position control method can be either open loop or closed loop.

4.2.4.2 Open loop position control

A device configured for open loop position control applies to a class of drive devices called stepper drives. This type of drive is illustrated in Figure 2.



IEC

Figure 2 – Open loop position control

A feedback device for this configuration is optional. In the absence of a feedback device, actual position can be estimated by the drive and returned to the controller.

4.2.4.3 Closed loop position control

A motor control device configured for closed loop position control is traditionally referred to as position loop drive or position servo drive. A position servo drive implies an inner velocity and torque control loop as shown in Figure 3. The presence of the torque/current control loop sometimes results in this kind of drive being referred to as a vector drive.

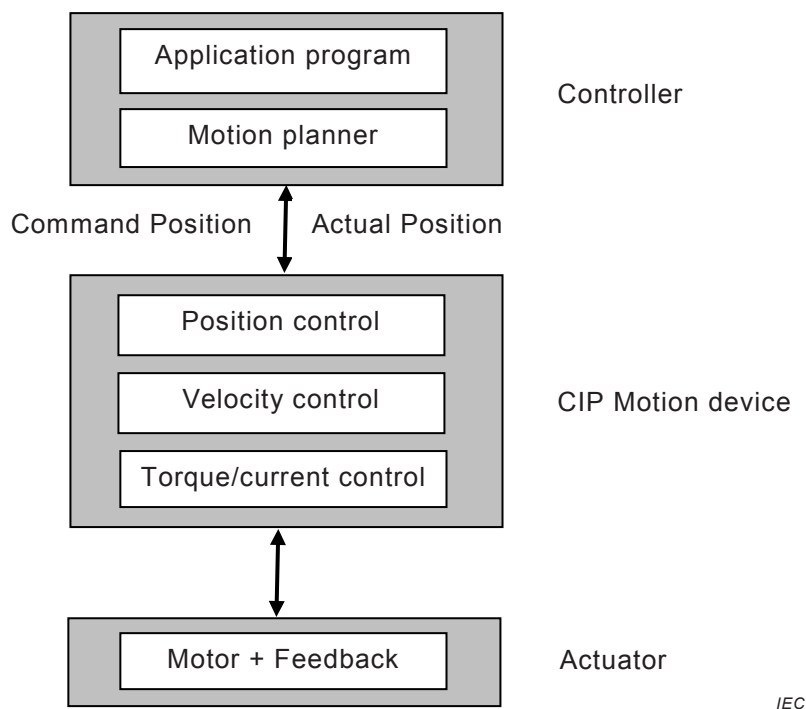


Figure 3 – Closed loop position control

A feedback device for this configuration is generally required to achieve good positioning accuracy. The feedback device may also be used to return Actual Velocity, and Actual Acceleration data to the controller via the cyclic data connection.

In addition to Command Position, the controller can pass Command Velocity and Command Acceleration for the purposes of forward control.

4.2.5 Velocity control

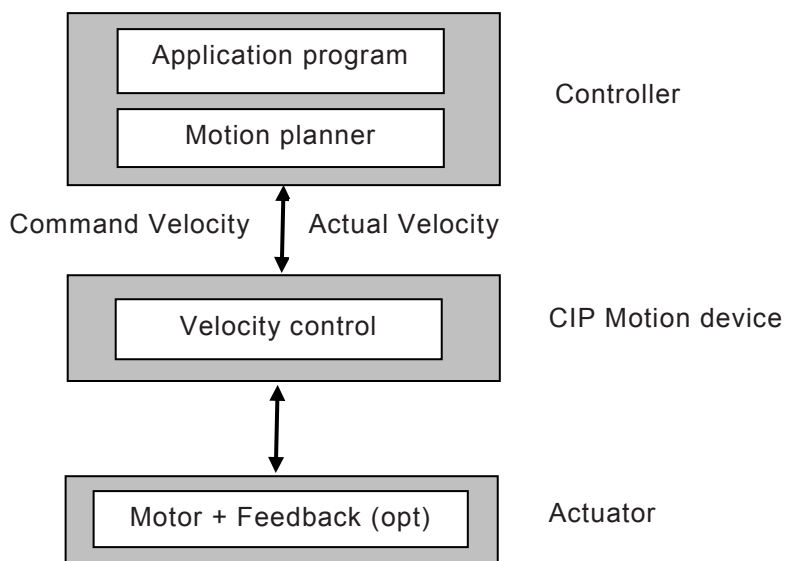
4.2.5.1 General concepts

In velocity control application mode, the application control program and motion planner provide a set-point value to the CIP Motion device via the cyclic data connection. The velocity control method can be either open loop or closed loop.

4.2.5.2 Open loop velocity control

A motor control device configured for open loop velocity control is traditionally referred to as variable frequency, or V/Hz, or VFD, drive. This type of drive is illustrated in Figure 4.

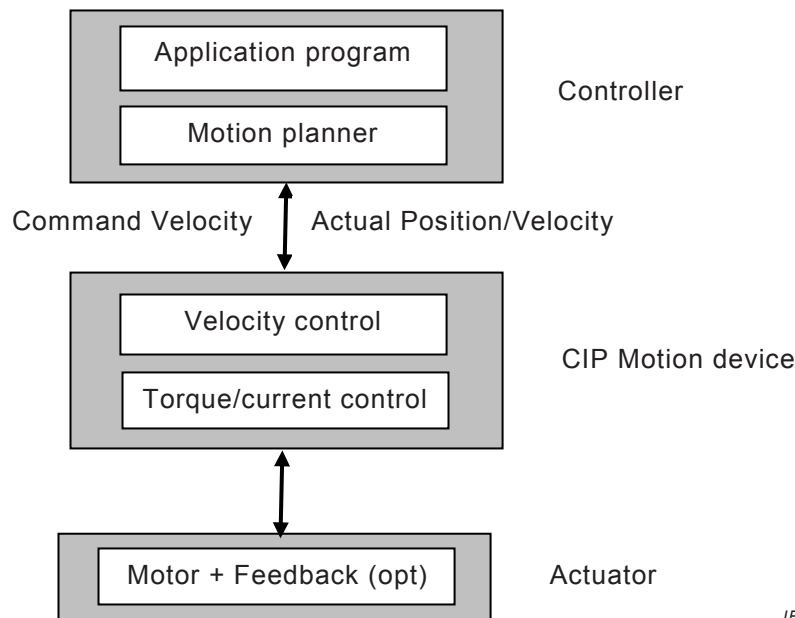
A feedback device for this configuration is optional. In the absence of a feedback device, actual velocity can be estimated by the drive and returned to the controller.



IEC

Figure 4 – Open loop velocity control**4.2.5.3 Closed loop velocity control**

A motor control device configured for closed loop velocity control is traditionally referred to as velocity loop drive or velocity servo drive. A closed loop velocity control drive implies an inner torque/current control loop (see Figure 5) and therefore is sometimes referred to as a vector drive.



IEC

Figure 5 – Closed loop velocity control

A feedback device for the velocity loop drive configuration is optional. Tighter speed regulation is achieved when using a feedback device, particularly at low speed. When the feedback device is included, it may be used to return actual position, velocity, and acceleration data to the controller via the cyclic data connection. When the feedback device is not included, only estimated velocity can typically be returned to the controller.

In addition to Command Velocity, the controller can also pass Command Acceleration for the purposes of forward control.

4.2.6 Acceleration control

While neither a mainstream control mode in the industry, nor mentioned in IEC 61800-7-1, the acceleration control mode is included here to complete the dynamic progression from velocity control to torque control and because the Motion Device Axis Object can support an Acceleration Command, potentially derived from the controller's motion planner. In the acceleration control mode, the application control program and motion planner provide acceleration set-point values to the CIP Motion device via the cyclic data connection. The drive converts the acceleration set-point into a torque command using the estimated system inertia. Acceleration control works in concert with the inner torque/current control loop as shown in Figure 6.

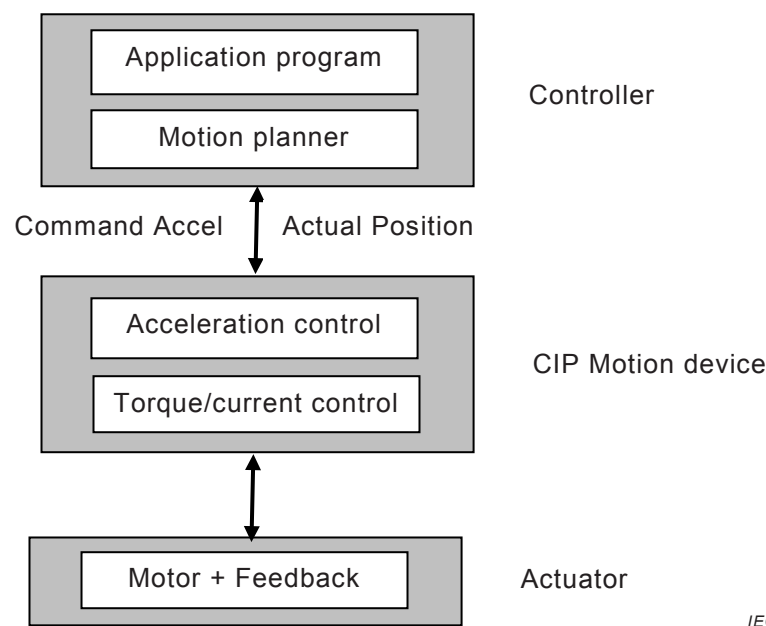
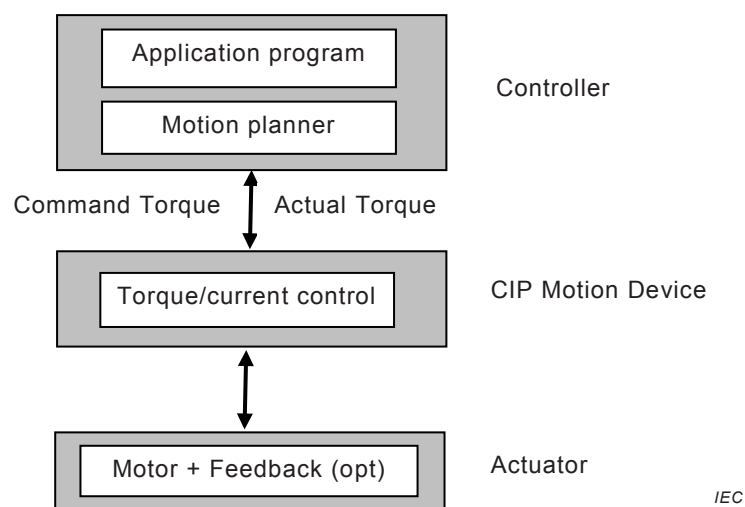


Figure 6 – Acceleration control

A feedback device for the acceleration control configuration is mandatory and may be used to return actual position, velocity, and acceleration data to the controller via the cyclic data connection.

4.2.7 Torque control

In torque control application mode, the application control program or the motion planner provide torque set-point values to the device via the cyclic data connection (see Figure 7). Because motor current and motor torque are generally related by a torque constant, K_t , torque control is often synonymous with current control.

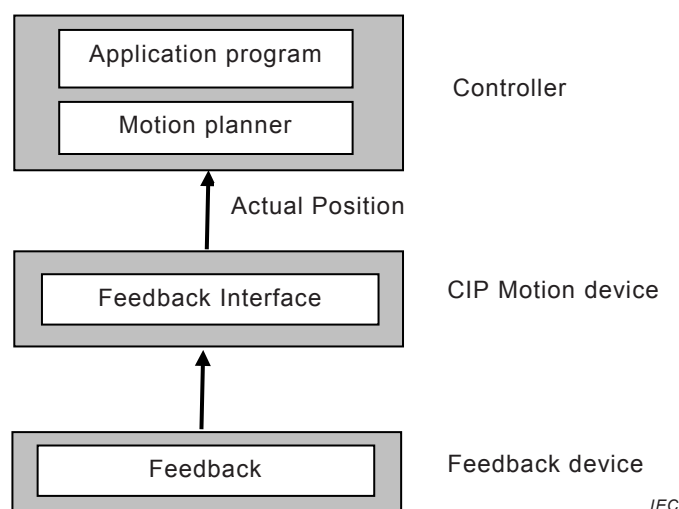
**Figure 7 – Torque control**

A position feedback device for this control mode is optional. If a feedback device is present, it may be used to return actual position, velocity, and acceleration data to the controller via the cyclic data connection.

4.2.8 No Control

The Motion Device Axis Object supports a “No Control” application mode where there is no dynamic motor control function. This mode is often used to support “feedback only” or “master feedback” functionality where a particular feedback channel in a CIP Motion drive device is serving as a master feedback source to the rest of the control system. This could also apply to integrated CIP Motion Encoder device types where the CIP Motion interface is applied directly to an Encoder.

In this “No Control” mode of operation, no set-point value is supplied to the CIP Motion device via the cyclic data connection, but actual position, velocity, and acceleration can be supplied by the device to the controller via the Cyclic Data Channel, if applicable. The No Control mode for Feedback Only functionality is illustrated in Figure 8.

**Figure 8 – No Control (Feedback Only)**

No Control mode also applies to other CIP Motion device types, such as standalone Bus Power Converters and dedicated motion I/O device types. Since there is no feedback channel associated with these device types, no actual position is returned to the controller.

5 Data types

5.1 Data type overview

Table 1 shows references of data types used in this profile and the related definitions.

Table 1 – Data types

Data types used in CIP Motion	Reference to definition
BOOL	Boolean (see 5.3.1.1.2 of IEC 61158-5-2:2014)
SINT	Integer8 (see 5.3.1.4.2.2 of IEC 61158-5-2:2014)
INT	Integer16 (see 5.3.1.4.2.4 of IEC 61158-5-2:2014)
DINT	Integer32 (see 5.3.1.4.2.6 of IEC 61158-5-2:2014)
LINT	Integer64 (see 5.3.1.4.2.8 of IEC 61158-5-2:2014)
USINT	Unsigned8 (see 5.3.1.4.3.2 of IEC 61158-5-2:2014)
UINT	Unsigned16 (see 5.3.1.4.3.4 of IEC 61158-5-2:2014)
UDINT	Unsigned32 (see 5.3.1.4.3.6 of IEC 61158-5-2:2014)
ULINT	Unsigned64 (see 5.3.1.4.3.8 of IEC 61158-5-2:2014)
REAL	Float32 (see 5.3.1.4.1.2 of IEC 61158-5-2:2014)
LREAL	Float64 (see 5.3.1.4.1.4 of IEC 61158-5-2:2014)
SWORD/BYTE	Bitstring8 (see 5.3.1.2.2 of IEC 61158-5-2:2014)
WORD	Bitstring16 (see 5.3.1.2.4 of IEC 61158-5-2:2014)
DWORD	Bitstring32 (see 5.3.1.2.6 of IEC 61158-5-2:2014)
LWORD	Bitstring64 (see 5.3.1.2.8 of IEC 61158-5-2:2014)

5.2 Conventions

Hexadecimal numbers in this part of the IEC 61800-7 series may be represented by nn_{hex} (if their exact data type is not defined) or $0xnn$, $0xn\text{nnnn}$ (if their data type is specified).

6 CIP Motion drive profile

6.1 Object model

6.1.1 Object overview

The object model in Figure 9 represents a CIP Motion device.

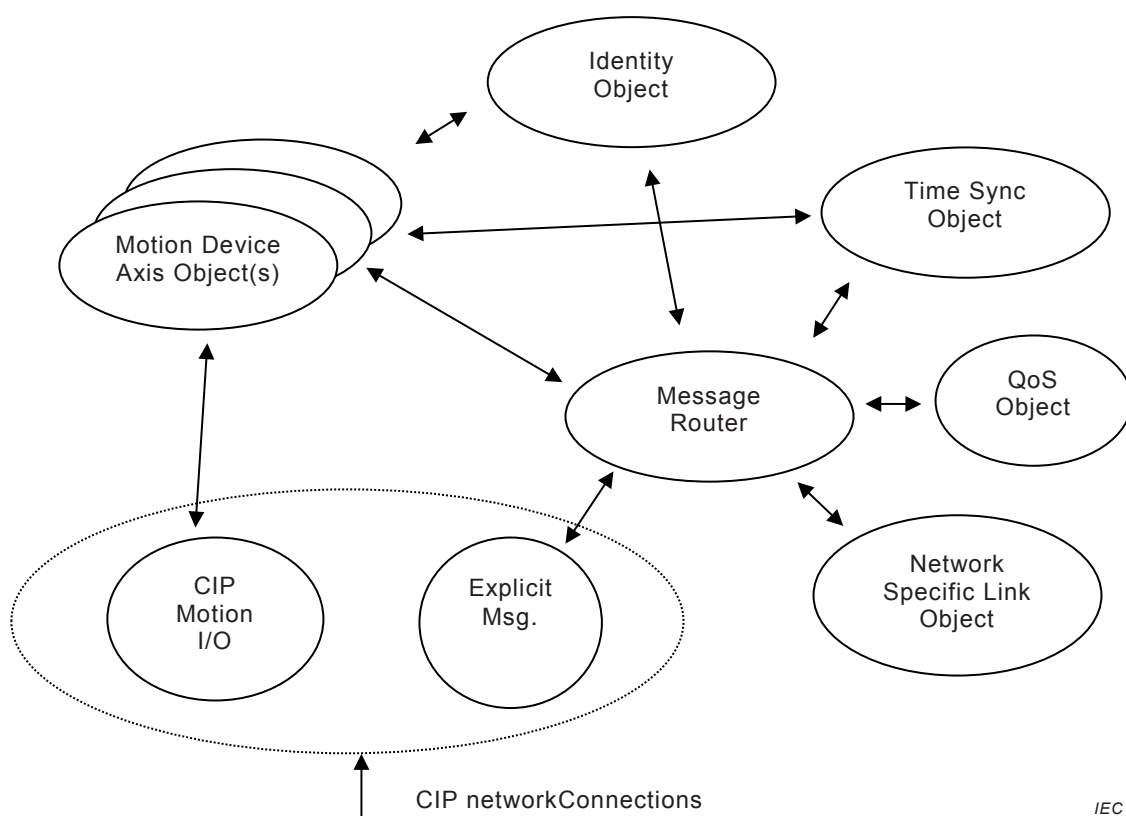


Figure 9 – Object Model for a CIP Motion device

Table 2 indicates:

- the object classes present in this device,
- whether or not the class is required,
- the number of instances present in each class.

Table 2 – Objects present in a CIP Motion device

Object class	Optional or required	Number of instances
CIP common required objects	Required	See IEC 61158-5-2 and IEC 61158-6-2
Motion Device Axis Object (0x42)	Required	1 per axis ^b
Time Sync Object (0x43)	Optional	1
QoS Object (0x48)	Conditional ^a	1
^a Required for EtherNet/IP		
^b An axis is an abstraction associated with a moving machine component, a converter power structure or a motion I/O device.		

Refer to IEC 61158-5-2, IEC 61158-6-2 and 6.4.6.1 for more details about these objects.

6.1.2 Object description

The object model in Figure 9 shows the main functional components of the CIP Motion device profile.

This object model also illustrates the use of multiple instances of a Motion Device Axis Object to implement a multi-axis motion device, such as a multi-axis drive. Each Motion Device Axis

Object instance governs the behavior of the associated axis. In this device profile, the term “axis” is synonymous with a “Motion Device Axis Object instance”. The implemented content of the Motion Device Axis Object instances is dictated by the specific CIP Motion Device Type according to Table 3.

Table 3 – Motion Device Axis Object content by Device Type

Device type	Motion Device Axis Object content
CIP Motion Drive	Support for one or more of F, P, V, T Device Function Codes
CIP Motion Encoder	Support for E, but no support for F, P, V, T Device Function Codes
CIP Motion Converter	Support for B Device Function Code only
CIP Motion I/O	Support for I/O Device Function Code only

A single bi-directional I/O connection to the Motion Device Axis Object class instance provides a cyclic data path between the controller and each individual Motion Device Axis Object instance. This connection passes on a special data structure whose self-defining format can be used to transfer cyclic, event, and service related data.

An optional Time Sync Object is included in the object model to facilitate accurate time synchronization between CIP Motion controllers and drive devices for high performance motion control. For lower performance drives such as V/Hz (VFD) drives, or simple velocity servo drives, the Time Sync Object is not necessary for drive operation.

A CIP Motion I/O Connection is a standard, cyclic, bidirectional CIP I/O connection whose packets can vary in size, based on the formats outlined in 6.4.2.2.

6.2 How objects affect behavior

The objects for this device affect the device’s behavior as shown in Table 4.

Table 4 – Object effect on behavior

Object class	Effect on behavior
CIP common required objects	See IEC 61158-5-2 and IEC 61158-6-2
Motion Device Axis Object	Provides dynamic control interface to drive, motor, and feedback components that comprise an axis.
Time Sync Object	Provide absolute time synchronization services between devices on the control network.
QoS Object	Provides configuration information associated with Ethernet QoS (Quality of Service) function.

6.3 Defining object interfaces

Objects supported for the CIP Motion device have the interfaces listed in Table 5.

Table 5 – Object interfaces

Object	Interface
CIP common required objects	See IEC 61158-5-2 and IEC 61158-6-2
Motion Device Axis Object	Message Router, CIP Motion I/O Connection, Time Sync Object, Identity Object
Time Sync Object	Message Router
QoS Object	Message Router

6.4 I/O connection messages

6.4.1 General

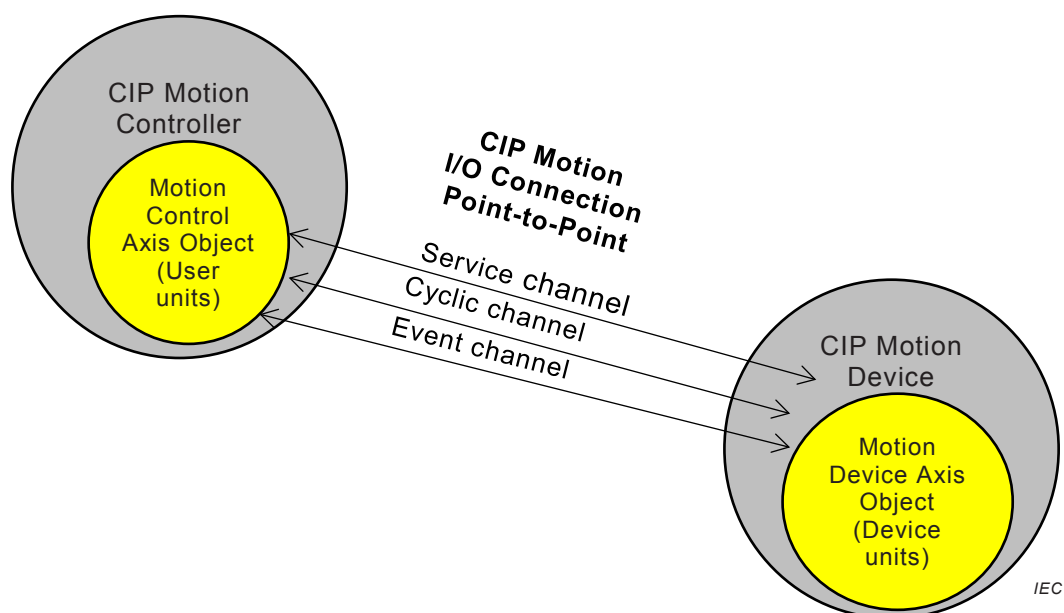
The CIP Motion device profile supports a Transport Class 1 point-to-point bi-directional I/O connection between the controller and the Motion Device Axis Object class: this I/O connection is specifically referred to as the CIP Motion I/O Connection.

The Motion Device Axis Object distributes the data in this connection to each instantiated Motion Device Axis Object instance.

6.4.2 CIP Motion I/O Connection

6.4.2.1 Overview

The following subclause specifies the CIP Motion I/O Connection format that includes the Controller-to-Device (C-to-D) Connection and the Device-to-Controller (D-to-C) Connection for bi-directional data transfer between a CIP Motion controller and a CIP Motion device, as shown in Figure 10.

**Figure 10 – CIP Motion I/O Connection model**

6.4.2.2 CIP Motion I/O Connection structure

Both CIP Motion I/O Connection data structures (Controller-to-Device and Device-to-Controller) begin with a Connection Header followed by a Time Data Block that typically includes a 64-bit time stamp. The Time Data Block is then followed by one or more Instance Data Blocks for each axis supported by the device node.

The size and contents of the Time Data Block and Instance Data Blocks vary dynamically as determined by the associated headers. This ability to vary the contents of the data blocks from update to update allows the device and controller to only send data that has changed from the last update, dramatically reducing the overall size of the typical CIP Motion Connection packet.

Each Instance Data Block within the CIP Motion I/O Connection packet consists of three sets of data blocks associated with the cyclic, event, and service data channels. From the device's perspective, these three distinct data channels have different data processing priorities as illustrated in Figure 11.

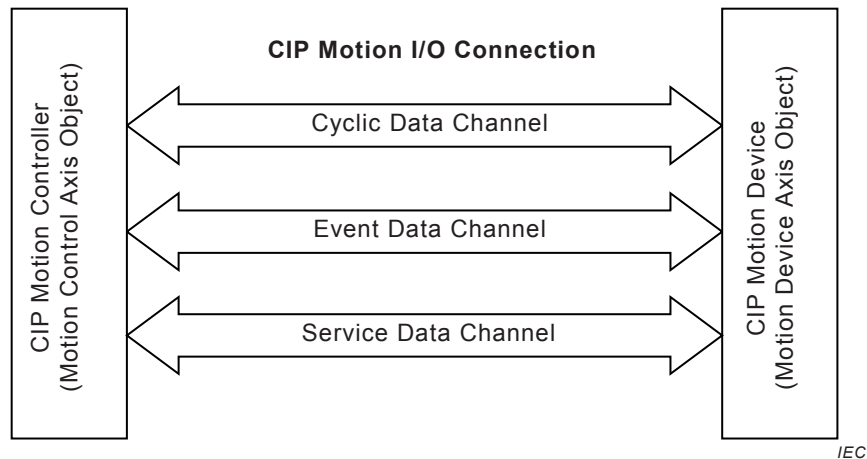


Figure 11 – CIP Motion I/O Connection channels

The specific functionality of these three data channels is as follows.

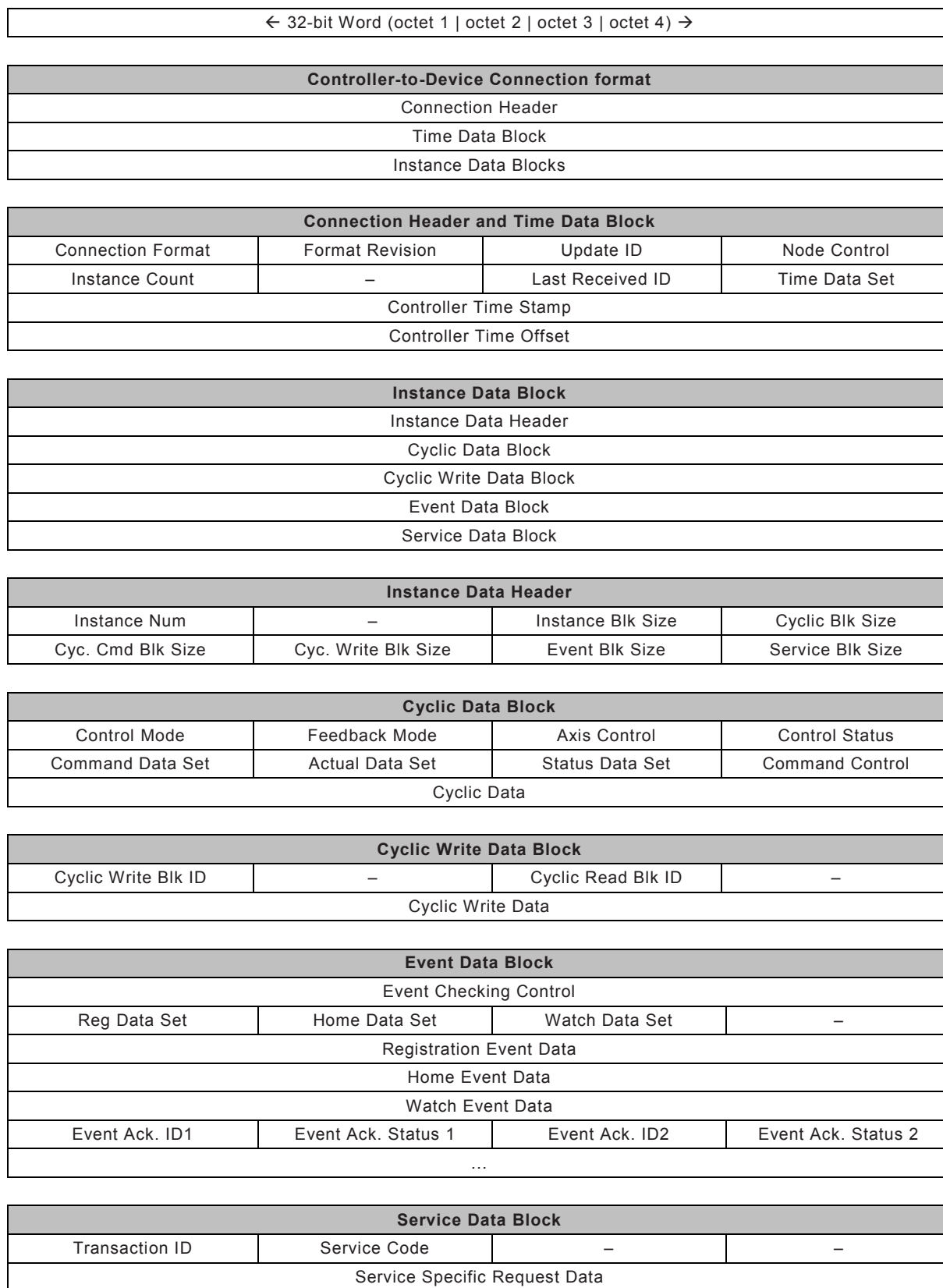
- **Cyclic Data Channel:** carries Cyclic Data Blocks that are sampled or calculated every Controller Update Period and synchronized with other nodes in the motion control system through use of distributed System Time. Cyclic data is high priority data that shall be immediately processed and applied to the device axis within one Device Update Period.
- **Event Data Channel:** carries event data associated with device event(s) (e.g. registration, homing, etc.) that have occurred within the last Controller Update Period. Event data is medium priority and shall be processed and applied within one Controller Update Period.
- **Service Data Channel:** carries data associated with service requests to read or write attribute values of the Motion Device Axis Object as part of on-line configuration and diagnostic functionality, as well as service requests to affect Motion Device Axis Object behavior as part of controller instruction execution. Service data has lowest priority and is typically buffered and processed as a background task. There is no guarantee that a service request will be processed within Controller Update Period.

Taken together, these three data channels provide a comprehensive controller to device data connection solution for industrial motion control.

6.4.2.3 I/O connection formats

An overview of the CIP Motion I/O Connection format is shown in Figure 12 and Figure 13.

Not shown in these format figures is the encapsulation associated with the Class 1 Transport that includes a Sequence Count. For a detailed description of the Class 1 Transport header refer to IEC 61158-5-2 and IEC 61158-6-2. Multi-octet data in the CIP I/O Connection data structure follows standard little-endian octet-addressing rule. In the figures, octet ordering reads left to right. The gray banners in the figures are section labels and are not part of the actual connection data structure. Unless otherwise stated, data structure elements defined as “reserved” or marked with a hyphen, “-”, shall be set to zero (0).



IEC

Figure 12 – Controller-to-Device Connection format (Connection Point 2)

← 32-bit Word (octet 1 octet 2 octet 3 octet 4) →			
Device-to-Controller Connection format			
Connection Header			
Time Data Block			
Instance Data Blocks			
Connection Header and Time Data Block			
Connection Format	Format Revision	Update ID	Node Status
Instance Count	Node Fault/Alarm	Last Received ID	Time Data Set
Device Time Stamp			
Device Time Offset			
Lost Updates	Late Updates	–	
Timing Diagnostic Data			
Instance Data Block			
Instance Data Header			
Cyclic Data Block			
Cyclic Read Data Block			
Event Data Block			
Service Data Block			
Instance Data Header			
Instance Num	–	Instance Blk Size	Cyclic Blk Size
Cyc. Act. Blk Size	Cyc. Read Blk Size	Event Blk Size	Service Blk Size
Cyclic Data Block			
Control Mode	Feedback Mode	Axis Response	Response Status
–	Actual Data Set	Status Data Set	Axis State
Cyclic Data			
Cyclic Read Data Block			
Cyclic Write Blk ID	Cyclic Write Status	Cyclic Read Blk ID	Cyclic Read Status
Cyclic Read Data			
Event Data Block			
Event Checking Status			
Reg Data Ack	Home Data Ack	Watch Data Ack	–
Event ID	Event Status	Event Type	–
Event Position			
Event Time Stamp			
...			
Service Data Block			
Transaction ID	Service Code	General Status	Extended Status
Service Specific Response Data			

IEC

Figure 13 – Device-to-Controller Connection format (Connection Point 2)

6.4.3 Controller-to-Device Connection

6.4.3.1 General

To facilitate a detailed description of each of its constituent data elements, the CIP Motion Controller-to-Device Connection is organised as shown in Figure 14.

CIP Motion Controller-to-Device Connection format
Connection Header
Time Data Block
Instance Data Blocks

IEC

Figure 14 – CIP Motion Controller-to-Device Connection format

6.4.3.2 Controller-to-Device Connection Header

6.4.3.2.1 General structure

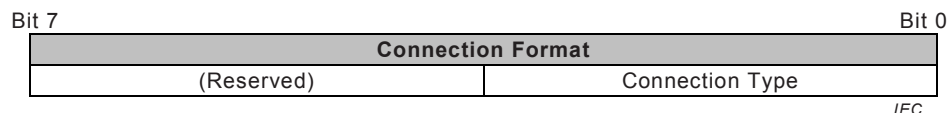
The Controller-to-Device Connection Header contains critical axis configuration information needed to parse the Instance Data Blocks. The fixed portion of the Connection Header is defined as shown in Figure 15.

Connection Header			
Connection Format	Format Revision	Update ID	Node Control

IEC

Figure 15 – Connection Header

- **Connection Format:** This byte determines the format of the CIP Motion Connection according to the definition in Figure 16.



IEC

Figure 16 – Connection Format

- **Connection Type:** This 4-bit value enumeration defines the CIP Motion Connection type as shown below. Valid values for a Controller-to-Device Connection are 2 for Fixed format and 6 for Variable format. Fixed connections have a fixed connection size during operation to support lower-performance CIP networks and are typically associated with a simple single axis device that does not support time synchronization services. Variable connections allow the connection data structure to vary in size during operation and are targeted for high-performance devices and CIP networks like EtherNet/IP.

- 0 = Fixed Controller Peer-to-Peer Connection
- 1 = Fixed Device Peer-to-Peer Connection
- 2 = Fixed Controller-to-Device Connection
- 3 = Fixed Device-to-Controller Connection
- 4 = Variable Controller Peer-to-Peer Connection
- 5 = Variable Device Peer-to-Peer Connection
- 6 = Variable Controller-to-Device Connection
- 7 = Variable Device-to-Controller Connection
- 8 to 15 = Reserved.

The Connection Format is not only used by the device to correctly parse the Controller-to-Device Connection data, but also by network diagnostic tools to display the individual elements of the data structure. If the device receives a Connection Format value that it cannot support, it generates a fault. In this case, the fault shall be Node Fault Error Code 4: Data Format Error. The Connection Format value shall not change for the life of the

connection and, therefore, need only be checked by the device for the first received packet.

- **Format Revision:** This edition of IEC 61800-7-202 defines Format Revision 2. Devices utilizing this edition of IEC 61800-7-202 shall only recognize Format Revision 2; Format Revision 1 is therefore rendered obsolete. This value is incremented by 1 for every revision of the Controller-to-Device Connection format that impacts the interface. The Format Revision allows newer devices to support the connection formats generated by an older controller. It also allows older devices to recognise a newer connection format from a controller that it cannot support and to generate a fault. In this case, the fault shall be Node Fault Error Code 4: Data Format Error. Network diagnostic tools can also key off the Format Revision to know how to parse the connection data packet. At the time the I/O Connection is established, the Format Revision number is also indicated by the I/O Connection Point in the Application Path of the Forward_Open service (see 6.4). The Format Revision value shall not change for the life of the connection and, therefore, need only be checked by the device for the first received packet.
- **Update ID:** This cyclic transaction number is incremented every update period. The Update ID is like the CIP sequence count and is used by the device to determine whether the connection buffer contains fresh data. In the case where the device is not synchronized, or does not support time synchronization services, the time stamp data is either not included or invalid, so the Update ID is the only way for the device to detect new connection data. In this case, the Controller-to-Device Connection Update ID is applied to the next Device-to-Controller Connection Update ID.

The Update ID also allows the device to determine if an update has been lost. Any skip in the Update ID is an indication that an update packet has been lost during transmission. If the number of consecutive missed updates exceeds the value given by the class attribute, Controller Update Delay High Limit, the Control Connection Update Fault is indicated by the Node Faults class attribute of the Motion Device Axis Object. This fault can only be cleared by closing and opening the CIP Motion I/O Connection. An optional Controller Update Delay Low Limit attribute is also available. If the number of consecutive missed updates exceeds the maximum value given by the class attribute, Controller Update Delay Low Limit, the Controller Connection Update Alarm is indicated by the Node Alarm class attribute of the Motion Device Axis Object. The alarm is cleared as soon as a fresh update is received.

- **Node Control:** This element is applied to the Motion Device Axis Object class attribute "Node Control", which is used to control the state of the associated device communications node. See 7.2.2.1 (Motion Device Axis Object) for details.

6.4.3.2.2 Fixed Connection Header

If the Connection Format is a Fixed Controller-to-Device Connection, the above header is immediately followed by the Instance Data Block.

6.4.3.2.3 Variable Connection Header

If the Connection Format is a Variable Controller-to-Device Connection, then the Connection Header contains additional fields related to multi-axis device addressing and time stamping, as shown in Figure 17.

Connection Header			
Connection Format	Format Revision	Update ID	Node Control
Instance Count	–	Last Received ID	Time Data Set
Controller Time Stamp			
Controller Time Offset			

IEC

Figure 17 – Connection Header

- **Instance Count:** This value reflects the number of Instance Data Blocks present in the CIP Motion Controller-to-Device Connection data structure.

- **Last Received ID:** This is the Update ID of the last Device-to-Controller Connection data block actually processed by the controller. The Last Received ID is used by the device to determine if the data that was sent was processed successfully. For example, Fault Codes passed as part of the Device-to-Controller Connections Status Data Block are sequenced into the controller's fault log. It is recommended that the Last Received ID element be read by the device to determine if the last Fault Code was entered into the Fault Log, thus allowing the device to send the next Fault Code in the sequence.
- **Time Data Set:** This bit-mapped byte contains flags that determine the usage and format of the controller and device timing information, as specified in Table 6. In general, the Time Data Set value sent by the controller in the Controller-to-Device Connection not only determines the contents of the Controller-to-Device Connection data block, but it also represents the requested Time Data Set to be sent by the device in the Device-to-Controller Connection. The value of this element is transferred to the Motion Device Axis Object attribute "Time Data Set".

Table 6 – Time Data Set

Bit	Definition	Syntax
0	Time Stamp	0 = no time stamp 1 = time stamp included
1	Time Offset	0 = no time offset – use last offset 1 = time offset included
2	Update Diagnostics	0 = no update diagnostic data in D-to-C 1 = update diagnostic data in D-to-C
3	Time Diagnostics	0 = no time diagnostic data in D-to-C 1 = time diagnostic data in D-to-C
4 to 7	(Reserved)	

- **Time Stamp** – if this bit is set, the Controller Time Stamp is included in the Time Data Set block. Setting this bit also instructs the device to send Device Time Stamps back to the controller in the Device-to-Controller Connection in the next update. It also instructs the device to send the Device Time Offset if its value has changed since the last update. If the device does not support time synchronization, the device simply returns a value of zero for the Time Stamp and does not ever return a Time Offset.
- **Time Offset** – if this bit is set, the Controller Time Offset is included in the Time Data Set block. This bit is only set when there is a change to the Controller Time Offset value, a relatively infrequent occurrence. Note that this bit does NOT determine if the device sends its Device Time Offset in the next update. That is determined by the Time Stamp bit and by whether or not the Device Time Offset has changed since the last update.
- **Update Diagnostics** – although there is no data associated with this bit in the Controller-to-Device Connection header, the Update Diagnostic bit in the Controller-to-Device Connection instructs the device to send Update Diagnostic data in the next Device-to-Controller Connection update, but only if one or more of the Update Diagnostic values have changed since the last update. This Update Diagnostic data includes the current number of Lost or Late Updates from the controller over the Controller-to-Device Connection. Update Diagnostics apply to devices regardless of the whether they support time synchronization.
- **Time Diagnostics** – although there is no data associated with this bit in the Controller-to-Device Connection header, the Time Diagnostic bit in the Controller-to-Device Connection instructs the device to send Time Diagnostic data in the next Device-to-Controller Connection update. This Time Diagnostic data includes the Controller-to-Device Received time stamp and the Device-to-Controller Data Transmit time stamp. If the device does not support time synchronization, the device simply returns a value of zero for the Time Diagnostic elements.
- **Controller Time Offset:** This element is included in the Connection Header if the Time Offset bit is set in the Time Data Set element. The Controller Time Offset represents the

64-bit System Time Offset value associated with the Controller Time Stamp that follows. The value of this element is transferred to the Motion Device Axis Object class attribute of the same name. The Controller Time Offset value is used by the device to determine if System Time as defined in the controller is skewed relative to System Time in the device. Normally, System Time between the device and the controller are closely matched at any given time. But every few seconds, according to the IEC 61588:2009 based CIP Sync protocol, the time master of the system can correct the System Time reference by a significant amount of time, perhaps an hour. Indeed, even the master itself can change as a higher quality time master is discovered by the system. These step changes to the System Time reference propagate through to the various devices on the network in such a way that the controller and the device are running with skewed System Time values. The Controller Time Offset value can be used together with the System Time Offset for the device to both determine if System Time between the device and the controller is skewed and if so, to compensate for the skew using the System Time Offset Compensation algorithm.

The Controller Time Offset value changes relatively infrequently, i.e. as a result of a IEC 61588 driven Sync update, which typically occurs every second. Thus, it is recommended that the controller only load and send the Time Offset data if the value has changed. The device need only parse the Controller Time Offset data if the associated Time Offset bit is set in the Time Data Set element, otherwise it continues to use the last Controller Time Offset received.

- **Controller Time Stamp:** This element is included in the Connection Header if the Time Stamp bit is set in the Time Data Set element. The Controller Time Stamp represents the 64-bit System Time value at the beginning of the Controller Update Period when the controller's update timer event occurred. In other words the Controller Time Stamp marks, in absolute time, the beginning of the current CIP Motion Connection cycle. The value of this time stamp is calculated by the controller as the sum of the controller's local clock value when update timer event occurred and the controller's System Time Offset value. The Controller Time Stamp is therefore directly associated with the command data contained in the connection. The value of this element, when read, is transferred to the Motion Device Axis Object class attribute Controller Time Stamp. Taken together with the Controller Update Period, established by the RPI of the Forward_Open service, and the Command Target Update (discussed later), the device has all the information it needs compute command interpolation polynomials to correct command data values for differences between the device and controller update timing. These differences can occur when the Controller Update Period is not an integer multiple of the Device Update Period or when the device updates are phase shifted relative to the controller.

When the connection is synchronized, the Controller Time Stamp can be used by the device along with the Controller Update Period to check for missed or late updates. If the difference between the last Controller Time Stamp and the current local Device Time Stamp exceeds the maximum value given by,

Max Fault Delay = Controller Update Delay High Limit × Controller Update Period,

the Control Connection Update Fault is indicated by the Node Faults class attribute of the Motion Device Axis Object. An optional Controller Update Delay Low Limit attribute is also available. If the difference between the last Controller Time Stamp and the current local Device Time Stamp exceeds the maximum value given by,

Max Alarm Delay = Controller Update Delay Low Limit × Controller Update Period,

the Controller Connection Update Alarm is indicated by the Node Alarm class attribute of the Motion Device Axis Object.

If either the Controller Time Offset or the Device Time Offset values have changed since the last update, System Time Offset Compensation shall be applied to the Controller Time Stamp. See 6.4.6.7 for details. If the time offsets have not changed, the typical case, the Controller Time Stamp can be directly applied to the Motion Device Axis Object.

6.4.3.3 Instance Data Blocks

6.4.3.3.1 General structure

After the Connection Header are one or more Instance Data Blocks as determined by the above Instance Count. The Instance Data Block has the basic structure shown in Figure 18.

Instance Data Block
Instance Data Header
Cyclic Data Block
Event Data Block
Service Data Block

IEC

Figure 18 – Instance Data Block

6.4.3.3.2 Instance Data Header

The Instance Data Header contains critical axis configuration information needed to parse and apply the data contained in the three data channels (see Figure 19). This header is only included in the Variable Connection format to accommodate multi-axis device applications. Information within the header can be used by the device communications interface to copy the individual data blocks into separate fixed memory locations for processing.

If configured for a Fixed Connection format, only the Cyclic Data Block for a single axis instance is supported so there is no need for any information to specify instance number or block sizing. The Instance Data Header is therefore not included.

Instance Data Header			
Instance Num	–	Instance Blk Size	Cyclic Blk Size
Cyc. Cmd. Blk Size	Cyc. Write Blk Size	Event Blk Size	Service Blk Size

IEC

Figure 19 – Instance Data Header

- **Instance Number:** This is the number that identifies the specific Motion Device Axis Object instance the following Instance Data Block applies to. Motion Device Axis Object instances are created as a contiguous series of instance numbers starting with instance 1. Within the connection data structure, the Instance Numbers for each consecutive Instance Data Block shall be an ordinal sequence, i.e. 0, 1, 2, 3, etc.

NOTE 1 Instance 0 is defined as the class instance and is typically only used during initialization to configure class attributes of the Motion Device Axis Object via the Service Data Block. Otherwise, the Instance Data Block sequence starts with instance 1.

Thus, in theory, up to 255 axis instances can be serviced by the CIP Motion I/O Connection. The instance numbers sequence of the Controller-to-Device Connection determines the instance number sequence of the subsequent Device-to-Controller Connection.

- **Instance Block Size:** This value represents the size of the Instance Data Block in 32-bit words including the header. The Instance Block Size is useful when the device wants to directly access the next Instance Data Block without having to add the sizes of the cyclic, event, and service blocks. If the Instance Block Size for the Controller-to-Device Connection is zero, i.e. no Instance Data Block, then the Instance Block Size for the following Device-to-Controller Connection shall also be zero.
- **Cyclic Block Size:** This value represents the size of the Cyclic Data Block in 32-bit word units including the header.

NOTE 2 For instance 0, the class instance, the Cyclic Block Size is 0 indicating that there is no Cyclic Data Block.

- **Cyclic Command Block Size:** This value represents the size of the Cyclic Command Data Block in 32-bit word units including the header. A Cyclic Command Block Size of 0 indicates that there is no Cyclic Command Data for the device to process.

- **Cyclic Write Block Size:** This value represents the size of the Cyclic Write Data Block in 32-bit word units including the header. A Cyclic Write Block Size of 0 indicates that there is no Cyclic Write data currently configured for transfer to the device and therefore there is no Cyclic Write Block included for the device to process.
- **Event Block Size:** This value represents the size of the Event Data Block in 32-bit word units including the header. If the Event Block Size for the Controller-to-Device Connection is zero, there is no active event checking, so the Event Block Size for the Device-to-Controller Connection shall also be zero. This effectively closes the Event Channel.
- **Service Block Size:** This value represents the size of the Service Data Block in 32-bit word units including the header. A Service Block Size value of 0 indicates there is no service request to process. If the Service Block Size for the Controller-to-Device Connection is zero, then the Service Block Size for the Device-to-Controller Connection shall also be zero. This effectively closes the Service Channel.

6.4.3.3.3 Cyclic Data Block

The Cyclic Data Block consists of high priority control data that is updated every connection cycle based on the Controller Update Period. The Cyclic Data Header at the top of the Cyclic Data Block is always included regardless of the connection format. This header contains key elements related to the content of the Cyclic Data Block of both the Controller-to-Device Connection and Device-to-Controller Connection, and the context of the data as determined by the Control Mode and Feedback Mode (see Figure 20). The header also provides a mechanism to control the state of the targeted device axis.

Cyclic Data Block			
Control Mode	Feedback Mode	Axis Control	Control Status
Command Data Set	Actual Data Set	Status Data Set	Command Control
Cyclic Data			

IEC

Figure 20 – Cyclic Data Block

- **Control Mode:** The lower 4-bits of this 8-bit element determine the control mode context of the command data as presently configured in the motion controller, as shown in Figure 21.

Bit 7	Control Mode		Bit 0
reserved – 4 bits		Control Mode – 4 bits	

IEC

Figure 21 – Control Mode

This value can be changed while on-line and even while the device axis is in the Running state. If a particular Control Mode transition is not supported by the device, an exception is generated that can be configured to perform any one of a number of actions in response to the illegal transition.

The value of this element is transferred to the Motion Device Axis Object attribute of the same name. The complete Control Mode attribute definition can be found in the attribute tables of the Motion Device Axis Object (see 7.3.2.2.1).

- **Feedback Mode:** The lower 4-bits of this 8-bit element determine the feedback context of the command data as presently configured in the motion controller, as shown in Figure 22.

Bit 7	Feedback Mode		Bit 0
reserved – 4 bits		Feedback Mode – 4 bits	

IEC

Figure 22 – Feedback Mode

Command position data can be referenced to feedback counts of either Feedback 1 or Feedback 2, or motor units for sensorless operation. This value can be changed while on-line and, if supported by the device, even while the device axis is in the Running state. If a particular Feedback Mode transition is not supported by the device, an exception is

generated that can be configured to perform any one of a number of actions in response to the illegal transition.

The value of this element is transferred to the Motion Device Axis Object attribute of the same name. The complete Feedback Mode attribute definition can be found in the attribute tables of the Motion Device Axis Object (see 7.3.6.2).

- **Axis Control:** An 8-bit enumerated code that can be used to directly execute Axis State change operations that do not require either passing or returning device parameters, and therefore, do not require a CIP service to initiate. This mechanism is fully described in the State Control subclause of the Motion Device Axis Object (see 7.6.1.2). Valid enumerations for this data element are shown in Table 7.

Table 7 – Axis Control

Request code	Requested operation
0	No Request
1	Enable Request
2	Disable Request
3	Shutdown Request
4	Shutdown Reset Request
5	Abort Request
6	Fault Reset Request
7	Stop Process Request
8	Change Actual Position Reference Request
9	Change Command Position Reference Request
10 to 126	(Reserved)
127	Cancel Request
128 to 255	(Vendor Specific)

- **Control Status:** A bit mapped value that can be interrogated by the device to determine the status of the control during the Initialization process. The value of this element is transferred to the optional Motion Device Axis Object attribute number 89, Control Status, if supported by the device. Valid bits for this data element are shown in Table 8.

Table 8 – Control Status

Bit	Control Status bit	Description
0	Configuration Complete	This bit is set when the controller has completed configuration of all axis instance attributes during Initialization phase.
1	(Reserved)	See Attribute 89 of the Motion Device Axis Object
2	Converter Bus Up	
3	Converter Bus Unload	
4	Converter AC Power Loss	
5 to 7	(Reserved)	

- **Command Data Set:** This bit mapped value has a bit defined for each possible real-time command reference (see Table 9). Command data appears in the same order in the Command Data Set as the bit numbers, so Command Acceleration would appear before Command Torque in the real-time data structure of the Controller-to-Device Connection.

The value of this element is transferred to the Motion Device Axis Object attribute "Command Data Set".

Table 9 – Command Data Set

Bit	Controller Command Data Element	Data type
0	Command Position	LREAL or DINT ^a
1	Command Velocity	REAL
2	Command Acceleration	REAL
3	Command Torque	REAL
4	(Reserved)	(Reserved)
5	(Reserved)	(Reserved)
6	Unwind Cycle Count	DINT
7	Position Displacement	DINT

^a Data Type for Command Position determined by Command Control element defined below. The Command Position data type can be either an LREAL or a DINT as determined by the Command Control element.

The above Controller Command Data Elements apply to the Controller-to-Device Connection's cyclic data structure and map to corresponding attributes in the Motion Device Axis Object as shown in Table 10.

The units of the Command Data Elements match the units defined for the associated Motion Device Axis Object attribute.

Table 10 – Command Data Element to Motion Device Axis Object attribute mapping

Bit	Command Data Element	Motion Device Axis Object attribute
0	Command Position (LREAL) Command Position (DINT)	Controller Position Command – Float Controller Position Command – Integer
1	Command Velocity	Controller Velocity Command
2	Command Acceleration	Controller Acceleration Command
3	Command Torque	Controller Torque Command
4	(Reserved)	(Reserved)
5	(Reserved)	(Reserved)
6	Unwind Cycle Count	Local variable
7	Position Displacement	Local variable

It is the job of the controller to ensure that the necessary Command Data Elements are included in the connection data to support the specified Control Mode.

- **Actual Data Set:** This bit mapped value has a bit defined for each possible real-time actual data attribute that is to be included in the Actual Data Set of the Device-to-Controller Connection's Instance Data Block in the next update (see Table 11). Actual data appears in the same order as the bit numbers, so Actual Position would appear before Actual Velocity in the Actual Data Set structure. Using this mechanism, the contents of the Actual Data Set may be changed at any time during device operation. The value of this element is transferred to the Motion Device Axis Object attribute "Actual Data Set".