

# **CiA Draft Standard Proposal 402**



## **Device Profile Drives and Motion Control**

This draft standard proposal is not recommended for implementation

**Version 2.0**

**Date: 17. March 2005**

**© CAN in Automation e.V.**

# HISTORY

Date	Changes
------	---------

July 2002	Document completely revised;
-----------	------------------------------

Summary of changes:

insert record definition 0080<sub>h</sub>: Interpolation time period record

insert record definition 0081<sub>h</sub>: Interpolation data configuration record

insert record definition 0082<sub>h</sub>: vl velocity acceleration deceleration record

object 6406<sub>h</sub>: change data type from DATE to TIME\_OF\_DAY

object 60C1<sub>h</sub>: change object code from RECORD to ARRAY

object 60F9<sub>h</sub>: change object code from RECORD to ARRAY

object 60FE<sub>h</sub>: change object code from RECORD to ARRAY

object 6089<sub>h</sub>: change category from optional to conditional

object 608A<sub>h</sub>: change category from optional to conditional

object 608B<sub>h</sub>: change category from optional to conditional

object 608C<sub>h</sub>: change category from optional to conditional

object 608D<sub>h</sub>: change category from optional to conditional

object 608E<sub>h</sub>: change category from optional to conditional

object 6084<sub>h</sub>: change category from mandatory to optional

object 606A<sub>h</sub>: change category from mandatory to optional

## General information on licensing and patents

CAN in AUTOMATION (CiA) calls attention to the possibility that some of the elements of this CiA specification may be subject of patent rights. CiA shall not be responsible for identifying any or all such patent rights.

### © CiA 2005-01-01

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from CiA at the address below.

CAN in Automation e. V.

Am Weichselgarten 26

DE - 91058 Erlangen, Germany

Tel.: +49-9131-69086-0

Fax: +49-9131-69086-79

Url: [www.can-cia.org](http://www.can-cia.org)

Email: [headquarters@can-cia.org](mailto:headquarters@can-cia.org)

# CONTENTS

<b>1</b>	<b>SCOPE.....</b>	<b>9</b>
<b>2</b>	<b>REFERENCES.....</b>	<b>10</b>
<b>3</b>	<b>DEFINITIONS AND ABBREVIATION.....</b>	<b>11</b>
<b>4</b>	<b>OVERVIEW .....</b>	<b>12</b>
4.1	Access to the drive.....	12
4.2	Architecture of the drive.....	13
<b>5</b>	<b>OPERATING PRINCIPLE.....</b>	<b>17</b>
5.1	Introduction .....	17
5.2	Standardization via profiling.....	17
5.3	The object dictionary.....	17
5.3.1	Index and sub-index usage.....	18
<b>6</b>	<b>EMERGENCY MESSAGES.....</b>	<b>19</b>
6.1	Principle.....	19
6.2	Error codes .....	19
<b>7</b>	<b>PREDEFINITIONS .....</b>	<b>24</b>
7.1	Predefined objects .....	24
7.1.1	Object 1000 <sub>h</sub> : <i>Device type</i> .....	24
7.1.2	Object 1001 <sub>h</sub> : <i>Error register</i> .....	24
7.1.3	Object 67FF <sub>h</sub> : <i>Single device type</i> .....	24
7.2	PDO mapping.....	24
7.2.1	Receive PDOs.....	25
7.2.2	Transmit PDOs.....	28
<b>8</b>	<b>OBJECT DICTIONARY .....</b>	<b>32</b>
<b>9</b>	<b>COMMON ENTRIES.....</b>	<b>33</b>
9.1	General information .....	33
9.1.1	Motor data.....	33
9.1.2	Drive data .....	33
9.2	Object dictionary entries.....	34
9.2.1	Objects defined in this chapter.....	34
9.3	Object description.....	34
9.3.1	Object 6007 <sub>h</sub> : <i>Abort connection option code</i> .....	34
9.3.2	Object 603F <sub>h</sub> : <i>Error code</i> .....	35
9.3.3	Object 6402 <sub>h</sub> : <i>Motor type</i> .....	35
9.3.4	Object 6403 <sub>h</sub> : <i>Motor catalog number</i> .....	36
9.3.5	Object 6404 <sub>h</sub> : <i>Motor manufacturer</i> .....	37
9.3.6	Object 6405 <sub>h</sub> : <i>http motor catalog address</i> .....	37
9.3.7	Object 6406 <sub>h</sub> : <i>Motor calibration date</i> .....	37
9.3.8	Object 6407 <sub>h</sub> : <i>Motor service period</i> .....	38
9.3.9	Object 6410 <sub>h</sub> : <i>Motor data</i> .....	38
9.3.10	Object 6502 <sub>h</sub> : <i>Supported drive modes</i> .....	39
9.3.11	Object 6503 <sub>h</sub> : <i>Drive catalog number</i> .....	40
9.3.12	Object 6504 <sub>h</sub> : <i>Drive manufacturer</i> .....	40
9.3.13	Object 6505 <sub>h</sub> : <i>http drive catalog address</i> .....	40

9.3.14	Object 6510 <sub>h</sub> : <i>Drive data</i> .....	41
9.3.15	Object 60FD <sub>h</sub> : <i>Digital inputs</i> .....	42
9.3.16	Object 60FE <sub>h</sub> : <i>Digital outputs</i> .....	42
<b>10</b>	<b>DEVICE CONTROL</b> .....	<b>44</b>
10.1	General information .....	44
10.1.1	State machine .....	45
10.2	Object dictionary entries .....	48
10.2.1	Objects defined in this chapter .....	48
10.3	Object description .....	49
10.3.1	Object 6040 <sub>h</sub> : <i>Controlword</i> .....	49
10.3.2	Object 6041 <sub>h</sub> : <i>Statusword</i> .....	50
10.3.3	Object 605B <sub>h</sub> : <i>Shutdown option code</i> .....	52
10.3.4	Object 605C <sub>h</sub> : <i>Disable operation option code</i> .....	53
10.3.5	Object 605A <sub>h</sub> : <i>Quick stop option code</i> .....	53
10.3.6	Object 605D <sub>h</sub> : <i>Halt option code</i> .....	54
10.3.7	Object 605E <sub>h</sub> : <i>Fault reaction option code</i> .....	55
10.3.8	Object 6060 <sub>h</sub> : <i>Modes of operation</i> .....	55
10.3.9	Object 6061 <sub>h</sub> : <i>Modes of operation display</i> .....	56
10.4	Functional description .....	57
10.4.1	Modes of operation function .....	57
10.4.2	Drive disabling function .....	58
10.4.3	Quick stop function .....	58
10.4.4	Stop function .....	58
10.4.5	Fault reaction .....	59
<b>11</b>	<b>FACTOR GROUP</b> .....	<b>60</b>
11.1	General information .....	60
11.1.1	Factors .....	60
11.1.2	Relationship between physical and internal units .....	60
11.2	Object dictionary entries .....	61
11.2.1	Objects defined in this chapter .....	61
11.3	Object description .....	61
11.3.1	Object 6089 <sub>h</sub> : <i>Position notation index</i> .....	61
11.3.2	Object 608A <sub>h</sub> : <i>Position dimension index</i> .....	62
11.3.3	Object 608B <sub>h</sub> : <i>Velocity notation index</i> .....	63
11.3.4	Object 608C <sub>h</sub> : <i>Velocity dimension index</i> .....	63
11.3.5	Object 608D <sub>h</sub> : <i>Acceleration notation index</i> .....	64
11.3.6	Object 608E <sub>h</sub> : <i>Acceleration dimension index</i> .....	64
11.3.7	Object 608F <sub>h</sub> : <i>Position encoder resolution</i> .....	65
11.3.8	Object 6090 <sub>h</sub> : <i>Velocity encoder resolution</i> .....	66
11.3.9	Object 6091 <sub>h</sub> : <i>Gear ratio</i> .....	67
11.3.10	Object 6092 <sub>h</sub> : <i>Feed constant</i> .....	68
11.3.11	Object 6093 <sub>h</sub> : <i>Position factor</i> .....	69
11.3.12	Object 6094 <sub>h</sub> : <i>Velocity encoder factor</i> .....	70
11.3.13	Object 6095 <sub>h</sub> : <i>Velocity factor 1</i> .....	71
11.3.14	Object 6096 <sub>h</sub> : <i>Velocity factor 2</i> .....	72
11.3.15	Object 6097 <sub>h</sub> : <i>Acceleration factor</i> .....	73
11.3.16	Object 607E <sub>h</sub> : <i>Polarity</i> .....	74
<b>12</b>	<b>PROFILE POSITION MODE</b> .....	<b>75</b>
12.1	General information .....	75

12.1.1	Input data description.....	76
12.1.2	Output data description.....	76
12.1.3	Internal states.....	76
12.2	Object dictionary entries.....	77
12.2.1	Objects defined in this chapter.....	77
12.2.2	Objects defined in other chapters.....	78
12.3	Object description.....	78
12.3.1	Object 607A <sub>h</sub> : <i>Target position</i> .....	78
12.3.2	Object 607B <sub>h</sub> : <i>Position range limit</i> .....	79
12.3.3	Object 607D <sub>h</sub> : <i>Software position limit</i> .....	80
12.3.4	Object 607F <sub>h</sub> : <i>Max profile velocity</i> .....	81
12.3.5	Object 6080 <sub>h</sub> : <i>Max motor speed</i> .....	81
12.3.6	Object 6081 <sub>h</sub> : <i>Profile velocity</i> .....	81
12.3.7	Object 6082 <sub>h</sub> : <i>End velocity</i> .....	82
12.3.8	Object 6083 <sub>h</sub> : <i>Profile acceleration</i> .....	82
12.3.9	Object 6084 <sub>h</sub> : <i>Profile deceleration</i> .....	83
12.3.10	Object 6085 <sub>h</sub> : <i>Quick stop deceleration</i> .....	83
12.3.11	Object 6086 <sub>h</sub> : <i>Motion profile type</i> .....	83
12.3.12	Object 60C5 <sub>h</sub> : <i>Max acceleration</i> .....	84
12.3.13	Object 60C6 <sub>h</sub> : <i>Max deceleration</i> .....	84
12.4	Functional description.....	85
<b>13</b>	<b>HOMING MODE.....</b>	<b>87</b>
13.1	General information.....	87
13.1.1	Input data description.....	87
13.1.2	Output data description.....	87
13.1.3	Internal states.....	87
13.2	Object dictionary entries.....	88
13.2.1	Objects defined in this chapter.....	88
13.2.2	Objects defined in other chapters.....	88
13.3	Object description.....	88
13.3.1	Object 607C <sub>h</sub> : <i>Home offset</i> .....	88
13.3.2	Object 6098 <sub>h</sub> : <i>Homing method</i> .....	89
13.3.3	Object 6099 <sub>h</sub> : <i>Homing speeds</i> .....	90
13.3.4	Object 609A <sub>h</sub> : <i>Homing acceleration</i> .....	91
13.4	Functional description.....	91
13.4.1	Homing methods.....	91
<b>14</b>	<b>POSITION CONTROL FUNCTION.....</b>	<b>96</b>
14.1	General information.....	96
14.1.1	Following error.....	96
14.1.2	Position reached.....	97
14.1.3	Input data description.....	97
14.1.4	Output data description.....	97
14.2	Object dictionary entries.....	97
14.2.1	Objects defined in this chapter.....	97
14.2.2	Objects defined in other chapters.....	98
14.3	Object description.....	98
14.3.1	Object 6062 <sub>h</sub> : <i>Position demand value</i> .....	98
14.3.2	Object 6063 <sub>h</sub> : <i>Position actual value*</i> .....	99
14.3.3	Object 6064 <sub>h</sub> : <i>Position actual value</i> .....	99

14.3.4	Object 6065 <sub>h</sub> : <i>Following error window</i> .....	99
14.3.5	Object 6066 <sub>h</sub> : <i>Following error time out</i> .....	100
14.3.6	Object 6067 <sub>h</sub> : <i>Position window</i> .....	100
14.3.7	Object 6068 <sub>h</sub> : <i>Position window time</i> .....	101
14.3.8	Object 60F4 <sub>h</sub> : <i>Following error actual value</i> .....	101
14.3.9	Object 60FA <sub>h</sub> : <i>Control effort</i> .....	102
14.3.10	Object 60FB <sub>h</sub> : <i>Position control parameter set</i> .....	102
14.3.11	Object 60FC <sub>h</sub> : <i>Position demand value*</i> .....	103
14.4	Functional description .....	104
<b>15</b>	<b>INTERPOLATED POSITION MODE.....</b>	<b>105</b>
15.1	General information .....	105
15.1.1	Input data description.....	107
15.1.2	Output data description .....	107
15.1.3	Internal states.....	108
15.2	Complex data types .....	109
15.2.1	Interpolation time period record.....	109
15.2.2	Interpolation data configuration record.....	110
15.3	Object dictionary entries .....	110
15.3.1	Objects defined in this chapter.....	110
15.3.2	Objects defined in other chapters.....	110
15.4	Object descriptions .....	111
15.4.1	Object 60C0 <sub>h</sub> : <i>Interpolation sub mode select</i> .....	111
15.4.2	Object 60C1 <sub>h</sub> : <i>Interpolation data record</i> .....	111
15.4.3	Object 60C2 <sub>h</sub> : <i>Interpolation time period</i> .....	113
15.4.4	Object 60C3 <sub>h</sub> : <i>Interpolation sync definition</i> .....	114
15.4.5	Object 60C4 <sub>h</sub> : <i>Interpolation data configuration</i> .....	115
15.5	Functional description .....	117
15.5.1	Interpolated position mode .....	117
15.5.2	Linear interpolated position mode with several axes .....	117
15.5.3	Buffer strategies for the interpolated position mode .....	118
<b>16</b>	<b>PROFILE VELOCITY MODE .....</b>	<b>120</b>
16.1	General Information .....	120
16.1.1	Input data description.....	123
16.1.2	Output data description .....	123
16.1.3	Internal states.....	123
16.2	Object dictionary entries .....	124
16.2.1	Objects defined in this chapter.....	124
16.2.2	Objects defined in other chapters.....	124
16.3	Object description .....	124
16.3.1	Object 6069 <sub>h</sub> : <i>Velocity sensor actual value</i> .....	125
16.3.2	Object 606A <sub>h</sub> : <i>Sensor selection code</i> .....	125
16.3.3	Object 606B <sub>h</sub> : <i>Velocity demand value</i> .....	126
16.3.4	Object 606C <sub>h</sub> : <i>Velocity actual value</i> .....	126
16.3.5	Object 606D <sub>h</sub> : <i>Velocity window</i> .....	126
16.3.6	Object 606E <sub>h</sub> : <i>Velocity window time</i> .....	127
16.3.7	Object 606F <sub>h</sub> : <i>Velocity threshold</i> .....	127
16.3.8	Object 6070 <sub>h</sub> : <i>Velocity threshold time</i> .....	128
16.3.9	Object 60FF <sub>h</sub> : <i>Target velocity</i> .....	128
16.3.10	Object 60F8 <sub>h</sub> : <i>Max slippage</i> .....	128

16.3.11	Object 60F9 <sub>h</sub> : <i>Velocity control parameter set</i> .....	129
16.4	Functional description .....	130
<b>17</b>	<b>PROFILE TORQUE MODE</b> .....	<b>131</b>
17.1	General information .....	131
17.1.1	Internal states.....	132
17.2	Object dictionary entries .....	133
17.2.1	Objects defined in this chapter.....	133
17.2.2	Objects defined in other chapters.....	133
17.3	Object description.....	134
17.3.1	Object 6071 <sub>h</sub> : <i>Target torque</i> .....	134
17.3.2	Object 6072 <sub>h</sub> : <i>Max torque</i> .....	134
17.3.3	Object 6073 <sub>h</sub> : <i>Max current</i> .....	134
17.3.4	Object 6074 <sub>h</sub> : <i>Torque demand value</i> .....	135
17.3.5	Object 6075 <sub>h</sub> : <i>Motor rated current</i> .....	135
17.3.6	Object 6076 <sub>h</sub> : <i>Motor rated torque</i> .....	136
17.3.7	Object 6077 <sub>h</sub> : <i>Torque actual value</i> .....	136
17.3.8	Object 6078 <sub>h</sub> : <i>Current actual value</i> .....	136
17.3.9	Object 6079 <sub>h</sub> : <i>DC link circuit voltage</i> .....	137
17.3.10	Object 6087 <sub>h</sub> : <i>Torque slope</i> .....	137
17.3.11	Object 6088 <sub>h</sub> : <i>Torque profile type</i> .....	138
17.3.12	Object 60F7 <sub>h</sub> : <i>Power stage parameters</i> .....	138
17.3.13	Object 60F6 <sub>h</sub> : <i>Torque control parameters</i> .....	139
<b>18</b>	<b>VELOCITY MODE</b> .....	<b>141</b>
18.1	General description.....	141
18.1.1	Input data description.....	141
18.1.2	Output data description .....	142
18.1.3	Structure of the velocity mode.....	142
18.1.4	Sub-function description.....	144
18.1.5	Internal states.....	144
18.2	Complex data types .....	146
18.2.1	vl velocity acceleration deceleration record.....	146
18.3	Object dictionary entries .....	146
18.3.1	Objects defined in this chapter.....	146
18.3.2	Objects defined in other chapters.....	147
18.4	Object description.....	147
18.4.1	Object 6042 <sub>h</sub> : <i>vl target velocity</i> .....	147
18.4.2	Object 6043 <sub>h</sub> : <i>vl velocity demand</i> .....	148
18.4.3	Object 6053 <sub>h</sub> : <i>vl percentage demand</i> .....	148
18.4.4	Object 6054 <sub>h</sub> : <i>vl actual percentage</i> .....	149
18.4.5	Object 6055 <sub>h</sub> : <i>vl manipulated percentage</i> .....	149
18.4.6	Object 604E <sub>h</sub> : <i>vl velocity reference</i> .....	149
18.4.7	Object 604C <sub>h</sub> : <i>vl dimension factor</i> .....	150
18.4.8	Object 604B <sub>h</sub> : <i>vl set-point factor</i> .....	152
18.4.9	Object 604D <sub>h</sub> : <i>vl pole number</i> .....	153
18.4.10	Object 6046 <sub>h</sub> : <i>vl velocity min max amount</i> .....	153
18.4.11	Object 6047 <sub>h</sub> : <i>vl velocity min max</i> .....	155
18.4.12	Object 6058 <sub>h</sub> : <i>vl frequency motor min max amount</i> .....	156
18.4.13	Object 6059 <sub>h</sub> : <i>vl frequency motor min max</i> .....	158
18.4.14	Object 6056 <sub>h</sub> : <i>vl velocity motor min max amount</i> .....	159

18.4.15	Object 6057 <sub>h</sub> : <i>vl velocity motor min max</i> .....	161
18.4.16	Object 6048 <sub>h</sub> : <i>vl velocity acceleration</i> .....	162
18.4.17	Object 6049 <sub>h</sub> : <i>vl velocity deceleration</i> .....	164
18.4.18	Object 604A <sub>h</sub> : <i>vl velocity quick stop</i> .....	165
18.4.19	Object 604F <sub>h</sub> : <i>vl ramp function time</i> .....	166
18.4.20	Object 6050 <sub>h</sub> : <i>vl slow down time</i> .....	167
18.4.21	Object 6051 <sub>h</sub> : <i>vl quick stop time</i> .....	168
18.4.22	Object 6044 <sub>h</sub> : <i>vl control effort</i> .....	168
18.4.23	Object 6045 <sub>h</sub> : <i>vl manipulated velocity</i> .....	169
18.4.24	Object 6052 <sub>h</sub> : <i>vl nominal percentage</i> .....	169
18.5	Functional description .....	170
18.5.1	Percentage function .....	170
18.5.2	Factor function and reverse factor function .....	170
18.5.3	Pole number function .....	171
18.5.4	Velocity limit function .....	171
18.5.5	Velocity motor limit function .....	172
18.5.6	Ramp function .....	172
18.5.7	Ramp min function .....	173
18.5.8	Reference calculation .....	173
18.5.9	Closed open loop control function .....	174
<b>19</b>	<b>APPENDIX</b> .....	<b>175</b>
19.1	Object dictionary by chapter .....	175
19.1.1	Common Entries .....	175
19.1.2	Device Control .....	176
19.1.3	Factor Group .....	176
19.1.4	Profile Position Mode .....	178
19.1.5	Homing Mode .....	179
19.1.6	Position control function .....	179
19.1.7	Interpolated position mode .....	180
19.1.8	Profile velocity mode .....	181
19.1.9	Profile Torque Mode .....	182
19.1.10	Velocity Mode .....	183
19.2	Object dictionary by index .....	186
19.3	Object dictionary by name .....	192
19.4	Definition of dimension indices .....	198
19.4.1	Dimension index table .....	198
19.4.2	Notation index table .....	199



# 1 SCOPE

This document represents the standardized CANopen device profile for digital controlled motion products like servo controllers, frequency converters or stepper motors.

All the devices mentioned above use communication techniques which conform to those described in the CiA Draft Standard DS 301 (CANopen Application Layer and Communication Profile). This document should be consulted in parallel to this profile.

## 2 REFERENCES

- /1/: ISO 7498, 1984, Information Processing Systems - Open Systems Interconnection - Basic Reference Model
- /2/: ISO 11898-1, 1999, Road Vehicles, Interchange of Digital Information - Controller Area Network (CAN) for high-speed Communication
- /3/: CiA DS 301, CANopen Application Layer and Communication Profile, Version 4.02, February 2002
- /4/: CiA DS 401, CANopen Device Profile I/O Modules, Version 2.1, May 2002
- /5/: DRIVECOM Profil Antriebstechnik/Profil 21
- /6/: DRIVECOM Profil Antriebstechnik/Servo 22, Jan. 1994

### 3 DEFINITIONS AND ABBREVIATION

<b>CAN</b>	<u>C</u> ontroller <u>A</u> rea <u>N</u> etwork
<b>CiA</b>	<u>C</u> AN in <u>A</u> utomation e. V.
<b>COB</b>	<u>C</u> ommunication <u>O</u> bject (CAN message). A unit of transportation in a CAN network. Data must be sent across a network inside a COB.
<b>COB-ID</b>	<u>C</u> OB- <u>I</u> dentifier. Identifies a COB uniquely in a network. The identifier determines the priority of that COB in the MAC sub-layer too.
<b>PDO</b>	<u>P</u> rocess <u>D</u> ata <u>O</u> bject. Object for data exchange between several devices.
<b>SDO</b>	<u>S</u> ervice <u>D</u> ata <u>O</u> bject. Peer to peer communication with access to the object dictionary of a device.
<b>pp</b>	Profile Position Mode
<b>pv</b>	Profile Velocity Mode
<b>vl</b>	Velocity Mode
<b>hm</b>	Homing Mode
<b>ip</b>	Interpolated Position Mode
<b>tq</b>	Profile Torque Mode
<b>all</b>	Mandatory for all modes
<b>ce</b>	Common entries in the object dictionary
<b>dc</b>	Device Control
<b>pc</b>	Position Control Function

## 4 OVERVIEW

### 4.1 Access to the drive

The access from the CAN network to the drive is done through data objects.

**Data objects of the drive**

<b>PDO</b>	<b>SDO</b>	<b>IDO</b>
<b>Process Data Object</b> described in chapters 9 to 18	<b>Service Data Object</b> described in chapter 7	<b>Internal Data Object</b> manufacturer specific normally not accessible

Figure 1: Data objects of the drive

**Process Data Object (PDO):** PDOs are messages in an unconfirmed service (see /3/). They are used for the transfer of real-time data to and from the drive. The transfer is fast, because it is performed with no protocol overhead what means to transport eight application data bytes in one CAN-frame. The PDOs correspond to entries in the object dictionary described in chapters 9 to 18. The data type and mapping of these objects into a PDO is described in chapter 7.

**Service Data Object (SDO):** SDOs are messages in a confirmed service with a kind of handshake (see /3/). They are used for the access to entries of the object dictionary. Especially the configuration for the requested behavior of the drive adapted to the various possible applications is done by these objects.

**Internal Data Object (IDO):** The internal data objects represent the adaptation of the manufacturer and device specific functionality to this profile. Normally these objects are not directly accessible; nevertheless a manufacturer can give the user access to the IDOs by SDO services.

## 4.2 Architecture of the drive

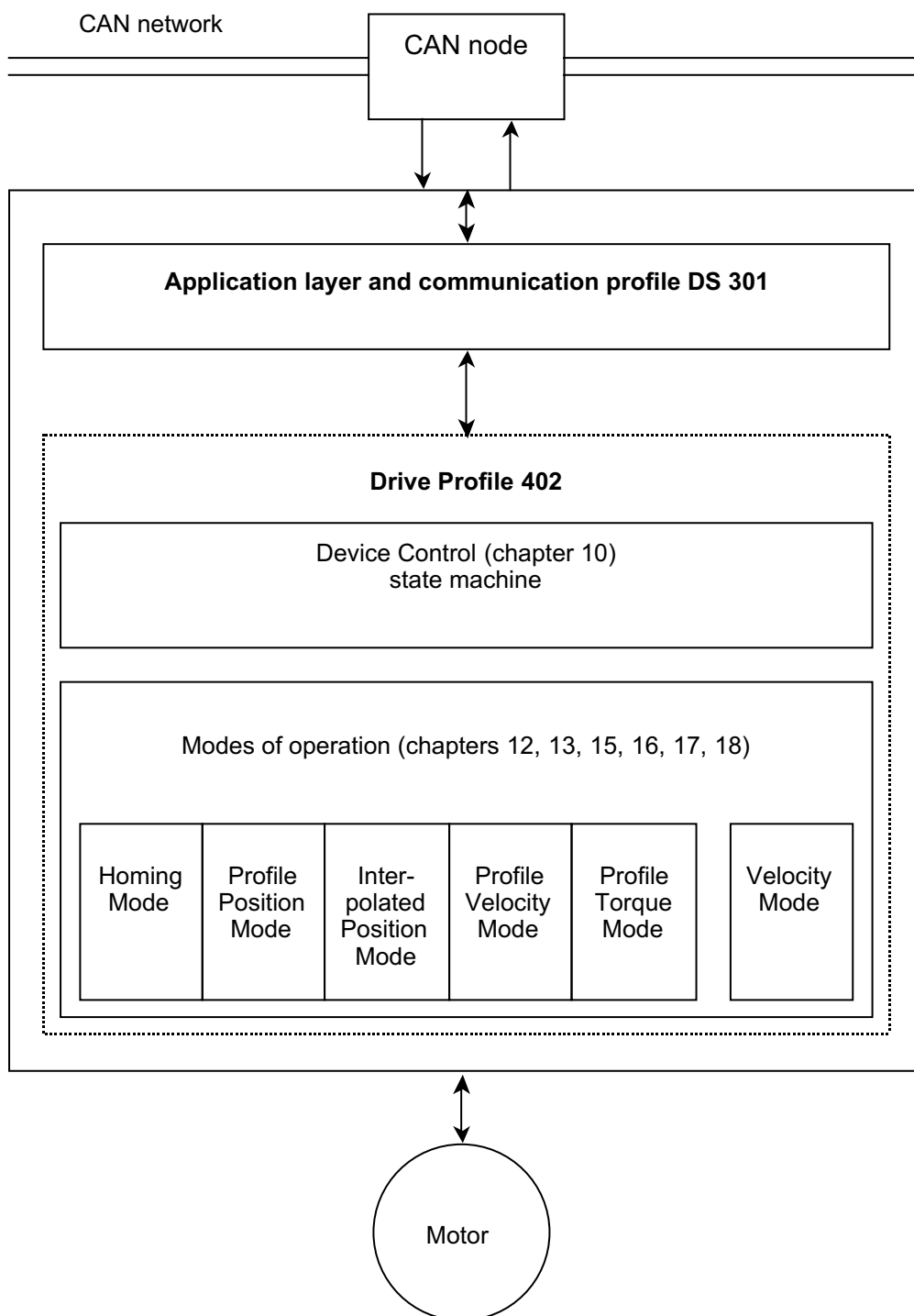


Figure 2: Communication architecture

**Device Control:** The starting and stopping of the drive and several mode specific commands are executed by the state machine. This is described in chapter 10. The mode specific actions are described in chapter 12 to 18.

**Modes of Operation:** The operation mode defines the behavior of the drive. The following modes are defined in this profile:

Homing mode (chapter 13)

This chapter describes the various methods to find a home position (also: reference point, datum, zero point).

Profile position mode (chapter 12)

The positioning of the drive is defined in this mode. Speed, position and acceleration can be limited and profiled moves using a Trajectory Generator are possible as well.

Interpolated position mode (chapter 15)

This chapter describes the time interpolation of single axes and the spatial interpolation of coordinated axes. Synchronization mechanisms and interpolation data buffers are covered by this chapter.

Profile velocity mode (chapter 16)

The Profile Velocity Mode is used to control the velocity of the drive with no special regard of the position. It supplies limit functions and Trajectory Generation.

Profile torque mode (chapter 17)

In this chapter the torque control with all related parameters is described.

Velocity mode (chapter 18)

Many frequency inverters use this simple mode to control the velocity of the drive with limits and ramp functions.

The velocity mode (chapter 18) is rather separated from the other modes and does not interfere with them so much. For this reason, the naming of object dictionary entries differs a little bit from the other chapters.

The manufacturer commits in the manual which modes are supported by his device.

If more than one mode is supported, then the manufacturer also defines whether the change of operation mode is allowed while the drive is moving or only when the drive is stopped.

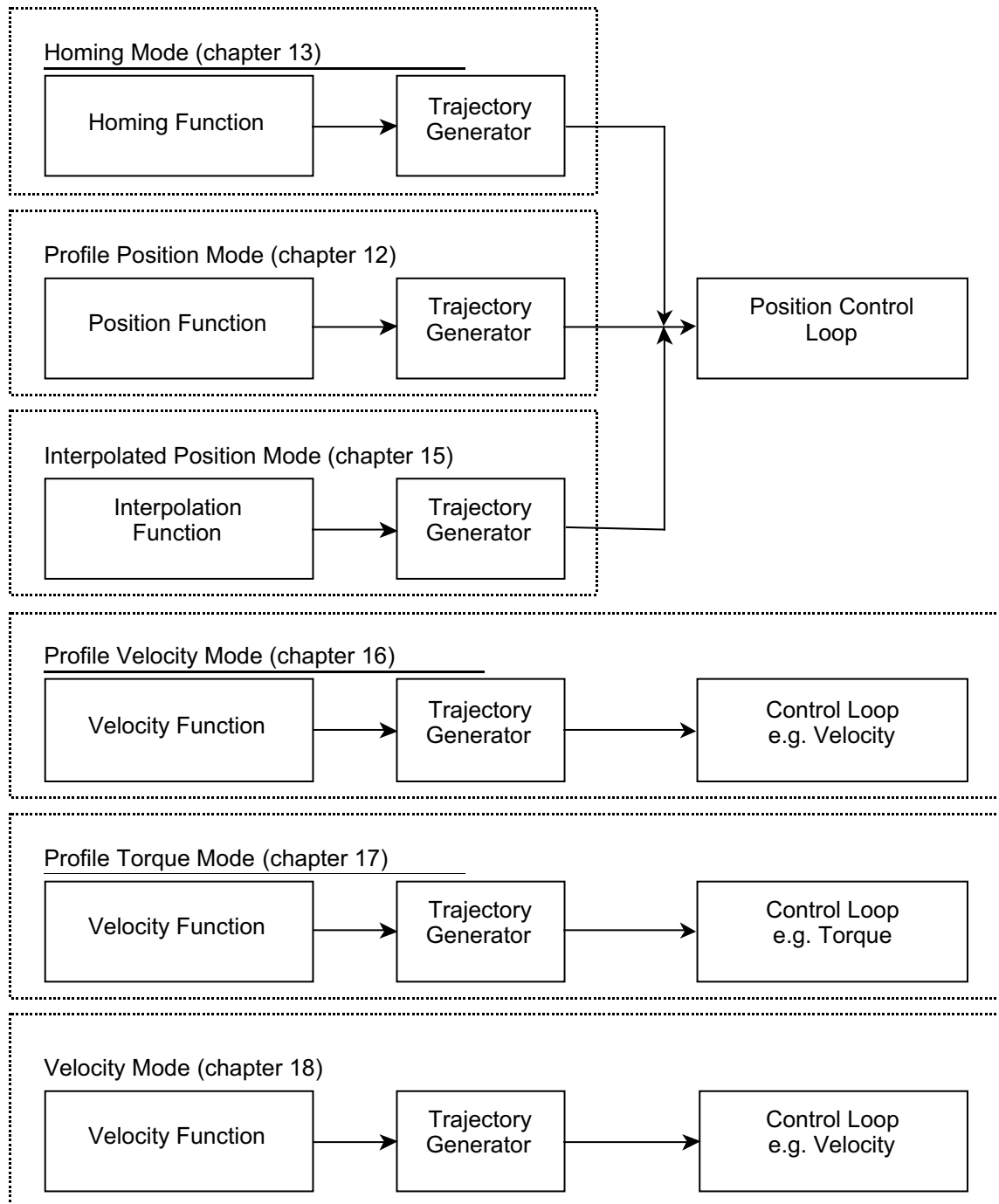


Figure 3: Functional architecture

**Trajectory generator:** The chosen operation mode and the corresponding parameters (objects) define the input of the trajectory generator. The trajectory generator supplies the control loop(s) with the demand values. They are generally mode specific.

Each mode may use its own trajectory generator. A general description of its functionality is given in chapter 12, which is related to the profile position mode.

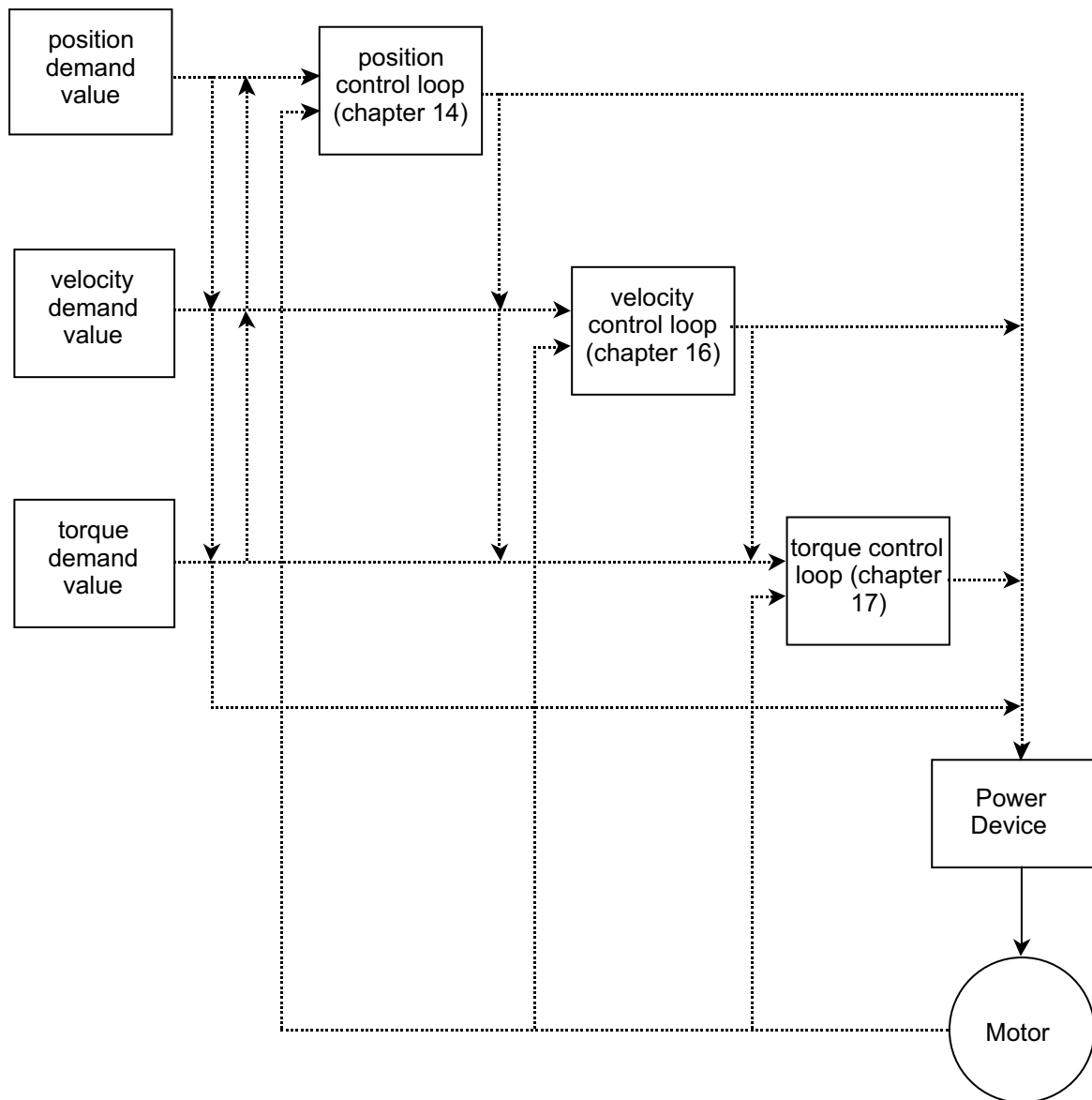


Figure 4: Possible structures of the control loop

**Control loop:** The implementation of the control loop is highly manufacturer specific and not described in this profile. Possible control loop structures are shown in the picture above.

The control loop can be open or closed and it can be operation mode specific or fixed. The objects which are described in chapter 12 to 18 must be implemented, if the corresponding mode is supported and if they are mandatory. But it is allowed that the manufacturer uses objects of the velocity controller in the profile position mode; for example the control loop structure consists of a position controller producing a velocity demand value and a velocity controller using this as a demand value.



## 5 OPERATING PRINCIPLE

### 5.1 Introduction

The purpose of this profile is, to give drives an understandable and unique behavior on the CAN network. The CANopen Device Profile for Drives and Motion Control is built on top of a CAN communication profile, called CANopen, describing the basic communication mechanisms common to all devices at the CAN-network.

The purpose of drive units is to connect axle controllers or other motion control products to the CAN bus. They can receive configuration information what is done via service data objects normally for I/O configurations, limit parameters for scaling or application specific parameters. At run time, data can be obtained from the drive unit via CAN bus by either polling or event driven (interrupt).

The motion control products have a process data object mapping for real time operation, which may be configured using service data objects (see /3/). This communication channel is used to interchange real-time data like set-points or actual values like a *position actual value* e.g.

### 5.2 Standardization via profiling

The two principal advantages of the profile approach for device specification are in the areas of system integration and device standardization.

If two independent device manufacturers design products that have to communicate, then both manufacturers must be provided with a device specification from the other one. These specifications will widely differ in formal and terminological aspects from one company to another. The concept of device profiling provides a standard for producing such specifications. By adopting this approach, all manufacturers will specify their devices in a similar fashion, what greatly reduces the effort involved in system integration.

The other obvious advantage of the profile approach for device specification is, that it can be used to guide manufacturers into producing standardized devices. The advantages of standardized devices are numerous. Perhaps most important is the idea, that a standardized device decouples a system integrator from a specific supplier. If one supplier cannot meet special application demands, a system designer can use devices from another supplier with reduced effort. On the other hand the device manufacturers are not forced any more to implement private protocols for each customer.

A device profile defines a 'standard' device. This standard device represents really basic functionality, every device within this device class must support. This mandatory functionality is necessary to ensure, that at least simple non-manufacturer-specific operation of a device is possible. For example the standard drive unit provides a 'Quick stop' function to stop a drive. This function is defined as mandatory, such that any drive unit supporting the CANopen Device Profile for Drives and Motion Control, can be halted using the same message.

The concept of device standardization is extended by the notion of optional functionality defined within the standardized device profile. Such optional functionality does not have to be implemented by all manufacturers. However, if a manufacturer implements such functionality he must do so in a fixed manner.

Providing optional functionality is a very powerful mechanism to ensure all manufacturers implementing particular functionality in a defined fashion. For example, the device profile covers multi-axes modules as well, which are still not very common. By defining a standardized access to the different axes, interchanging devices from different manufacturers becomes easier.

The device profiles provide a mechanism by which manufacturers wishing to implement truly manufacturer specific functionality can do so as well. This is clearly necessary since it would be impossible to anticipate all possible device functionality and define this in the optional category of each device class. This concept guarantees that the standard device profiles are 'future-proof'.

By defining mandatory device characteristics, basic network operation is guaranteed. By defining optional device features a degree of defined flexibility can be built in. By leaving 'hooks' for manufacturer specific functionality, manufacturers will not be constrained to an out-of-date standard.

### 5.3 The object dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index so that the object dictionary may contain a maximum of 65536 entries.

The layout closely conforms with device profiles for other field bus systems and is described in detail in /3/.

The standardized device profile area at indices 6000<sub>h</sub> through 9FFF<sub>h</sub> contains all data objects common to a class of devices that can be read or written via the network. The drives profile uses entries from 6000<sub>h</sub> to 9FFF<sub>h</sub> to describe the drive parameters and the drive functionality. Within this range up to 8 axles can be realized. Additionally it is possible to describe optional I/O modules combined with the drive. These I/O modules must conform to DS 401 (see /4/) and can be implemented instead of an axle. For standard drives only the range 6000<sub>h</sub> to 67FF<sub>h</sub> is mandatory. There are also two reserved areas at indices 060<sub>h</sub> through 0FFF<sub>h</sub> and A000<sub>h</sub> through FFFF<sub>h</sub> for future use by the communication or drive profile.

For multi axles devices the object range 6000<sub>h</sub> to 67FF<sub>h</sub> is shifted as follows:

6000<sub>h</sub> to 67FF<sub>h</sub> axle 0  
 6800<sub>h</sub> to 6FFF<sub>h</sub> axle 1  
 7000<sub>h</sub> to 77FF<sub>h</sub> axle 2  
 7800<sub>h</sub> to 7FFF<sub>h</sub> axle 3  
 8000<sub>h</sub> to 87FF<sub>h</sub> axle 4  
 8800<sub>h</sub> to 8FFF<sub>h</sub> axle 5  
 9000<sub>h</sub> to 97FF<sub>h</sub> axle 6  
 9800<sub>h</sub> to 9FFF<sub>h</sub> axle 7

### 5.3.1 Index and sub-index usage

A 16-bit index is used to address all entries within the object dictionary. In case of a simple variable this references the value of this variable directly. In case of records and arrays however, the index addresses the whole data structure. To allow individual elements of structures of data to be accessed via the network a sub-index has been defined. For single object dictionary entries such as an Unsigned8, Boolean, Integer32 etc. the value for the sub-index is always zero. For complex object dictionary entries such as arrays or records with multiple data fields the sub-index refers to fields within a data-structure pointed to by the main index. Index counting starts with one. For example in the chapter Factor Group exists the object 608F<sub>h</sub> named *position encoder resolution*. Because this may be a fraction, two integers in an array are used to describe it. The drive uses the two values in the following manner:

$$\text{position encoder resolution} = \frac{\text{encoder increments}}{\text{motor revolutions}}$$

The sub-index concept can be used to access these individual fields which may be of different data type as shown below:

Index	Sub	Name	Data type
648F <sub>h</sub>	0	Number of elements	UNSIGNED8
	1	Encoder increments	UNSIGNED32
	2	Motor revolutions	UNSIGNED32

Table 1: Usage of index and sub-index

## 6 EMERGENCY MESSAGES

### 6.1 Principle

Emergency messages are triggered by internal errors in the device and they are assigned the highest possible priority to ensure that they get access to the bus without latency. The Emergency Messages contain an error field with pre-defined error codes and additional information (see /3/).

Error codes from xx00<sub>h</sub> to xx7F<sub>h</sub> are defined in /3/ or in this profile. Not defined error codes within this range are reserved. Error codes between xx80<sub>h</sub> and xxFF<sub>h</sub> can be used manufacturer specific.

After initialization the device has to send emergency messages in the error case.

### 6.2 Error codes

Error code (hex)	Meaning	Defined by
0000	No error	DS 301
1000	Generic error	DS 301
2000	Current	DS 301
2100	Current on device input side	DS 301
2110	Short circuit/earth leakage	DS 402
2120	Earth leakage	DS 402
2121	Earth leakage phase L1	DS 402
2122	Earth leakage phase L2	DS 402
2123	Earth leakage phase L3	DS 402
2130	Short circuit	DS 402
2131	Short circuit phases L1-L2	DS 402
2132	Short circuit phases L2-L3	DS 402
2133	Short circuit phases L3-L1	DS 402
2200	Internal current	DS 301
2211	Internal current No.1	DS 402
2212	Internal current No.2	DS 402
2213	Over-current in ramp function	DS 402
2214	Over-current in the sequence	DS 402
2220	Continuous over current	DS 402
2221	Continuous over current No.1	DS 402
2222	Continuous over current No.2	DS 402
2230	Short circuit/earth leakage	DS 402
2240	Earth leakage	DS 402
2250	Short circuit	DS 402
2300	Current on device output side	DS 301
2310	Continuous over current	DS 402
2311	Continuous over current No.1	DS 402
2312	Continuous over current No.2	DS 402
2320	Short circuit/earth leakage	DS 402
2330	Earth leakage	DS 402
2331	Earth leakage phase U	DS 402
2332	Earth leakage phase V	DS 402
2333	Earth leakage phase W	DS 402
2340	Short circuit	DS 402
2341	Short circuit phases U-V	DS 402
2342	Earth leakage phase V-W	DS 402
2343	Earth leakage phase W-U	DS 402

Error code (hex)	Meaning	Defined by
3000	Voltage	DS 301
3100	Mains voltage	DS 301
3110	Mains over-voltage	DS 402
3111	Mains over-voltage phase L1	DS 402
3112	Mains over-voltage phase L2	DS 402
3113	Mains over-voltage phase L3	DS 402
3120	Mains under-voltage	DS 402
3121	Mains under-voltage phase L1	DS 402
3122	Mains under-voltage phase L2	DS 402
3123	Mains under-voltage phase L3	DS 402
3130	Phase failure	DS 402
3131	Phase failure L1	DS 402
3132	Phase failure L2	DS 402
3133	Phase failure L3	DS 402
3134	Phase sequence	DS 402
3140	Mains frequency	DS 402
3141	Mains frequency too great	DS 402
3142	Mains frequency too small	DS 402
3200	DC link voltage	DS 301
3210	DC link over-voltage	DS 402
3211	Over-voltage No. 1	DS 402
3212	Over voltage No. 2	DS 402
3220	DC link under-voltage	DS 402
3221	Under-voltage No. 1	DS 402
3222	Under-voltage No. 2	DS 402
3230	Load error	DS 402
3300	Output voltage	DS 301
3310	Output over-voltage	DS 402
3311	Output over-voltage phase U	DS 402
3312	Output over-voltage phase V	DS 402
3313	Output over-voltage phase W	DS 402
3320	Armature circuit	DS 402
3321	Armature circuit interrupted	DS 402
3330	Field circuit	DS 402
3331	Field circuit interrupted	DS 402
4000	Temperature	DS 301
4100	Ambient temperature	DS 301
4110	Excess ambient temperature	DS 402
4120	Too low ambient temperature	DS 402
4130	Temperature supply air	DS 402
4140	Temperature air outlet	DS 402
4200	Temperature device	DS 301
4210	Excess temperature device	DS 402
4220	Too low temperature device	DS 402
4300	Temperature drive	DS 402
4310	Excess temperature drive	DS 402
4320	Too low temperature drive	DS 402
4400	Temperature supply	DS 402
4410	Excess temperature supply	DS 402

Error code (hex)	Meaning	Defined by
4420	Too low temperature supply	DS 402
5000	Device hardware	DS 301
5100	Supply	DS 402
5110	Supply low voltage	DS 402
5111	U1 = supply +/- 15V	DS 402
5112	U2 = supply +24 V	DS 402
5113	U3 = supply +5 V	DS 402
5114	U4 = manufacturer specific	DS 402
5115	U5 = manufacturer specific	DS 402
5116	U6 = manufacturer specific	DS 402
5117	U7 = manufacturer specific	DS 402
5118	U8 = manufacturer specific	DS 402
5119	U9 = manufacturer specific	DS 402
5120	Supply intermediate circuit	DS 402
5200	Control	DS 402
5210	Measurement circuit	DS 402
5220	Computing circuit	DS 402
5300	Operating unit	DS 402
5400	Power section	DS 402
5410	Output stages	DS 402
5420	Chopper	DS 402
5430	Input stages	DS 402
5440	Contacts	DS 402
5441	Contact 1 = manufacturer specific	DS 402
5442	Contact 2 = manufacturer specific	DS 402
5443	Contact 3 = manufacturer specific	DS 402
5444	Contact 4 = manufacturer specific	DS 402
5445	Contact 5 = manufacturer specific	DS 402
5450	Fuses	DS 402
5451	S1 = I1	DS 402
5452	S2 = I2	DS 402
5453	S3 = I3	DS 402
5454	S4 = manufacturer specific	DS 402
5455	S5 = manufacturer specific	DS 402
5456	S6 = manufacturer specific	DS 402
5457	S7 = manufacturer specific	DS 402
5458	S8 = manufacturer specific	DS 402
5459	S9 = manufacturer specific	DS 402
5500	Data storage	DS 402
5510	Working memory	DS 402
5520	Program memory	DS 402
5530	Non-volatile data memory	DS 402
6000	Device software	DS 301
6010	Software reset (watchdog)	DS 402
6100	Internal software	DS 301
6200	User software	DS 301
6300	Data record	DS 301
6301	Data record No. 1	DS 402
	.....	

Error code (hex)	Meaning	Defined by
630F	Date record No.15	DS 402
6310	Loss of parameters	DS 402
6320	Parameter error	DS 402
7000	Additional modules	DS 301
7100	Power	DS 402
7110	Brake chopper	DS 402
7111	Failure brake chopper	DS 402
7112	Over current brake chopper	DS 402
7113	Protective circuit brake chopper	DS 402
7120	Motor	DS 402
7121	Motor blocked	DS 402
7122	Motor error or commutation malfunc.	DS 402
7123	Motor tilted	DS 402
7200	Measurement circuit	DS 402
7300	Sensor	DS 402
7301	Tacho fault	DS 402
7302	Tacho wrong polarity	DS 402
7303	Resolver 1 fault	DS 402
7304	Resolver 2 fault	DS 402
7305	Incremental sensor 1 fault	DS 402
7306	Incremental sensor 2 fault	DS 402
7307	Incremental sensor 3 fault	DS 402
7310	Speed	DS 402
7320	Position	DS 402
7400	Computation circuit	DS 402
7500	Communication	DS 402
7510	Serial interface No. 1	DS 402
7520	Serial interface No. 2	DS 402
7600	Data storage	DS 402
8000	Monitoring	DS 301
8100	Communication	DS 301
8110	CAN overrun (objects lost)	DS 301
8120	CAN in Error Passive Mode	DS 301
8130	Life guard error or heartbeat error	DS 301
8140	Recovered from bus-off	DS 301
8150	Transmit COB-ID	DS 301
8200	Protocol error	DS 301
8210	PDO not processed due to length error	DS 301
8220	PDO length exceeded	DS 301
8300	Torque control	DS 402
8311	Excess torque	DS 402
8312	Difficult start up	DS 402
8313	Standstill torque	DS 402
8321	Insufficient torque	DS 402
8331	Torque fault	DS 402
8400	Velocity speed controller	DS 402
8500	Position controller	DS 402
8600	Positioning controller	DS 402
8611	Following error	DS 402

Error code (hex)	Meaning	Defined by
8612	Reference limit	DS 402
8700	Sync controller	DS 402
8800	Winding controller	DS 402
8900	Process data monitoring	DS 402
8A00	Control	DS 402
9000	External error	DS 301
F000	Additional functions	DS 301
F001	Deceleration	DS 402
F002	Sub-synchronous run	DS 402
F003	Stroke operation	DS 402
F004	Control	DS 402
FF00	Manufacturer specific	
	.....	
FFFF	Manufacturer specific	

Table 2: Error codes

## 7 PREDEFINITIONS

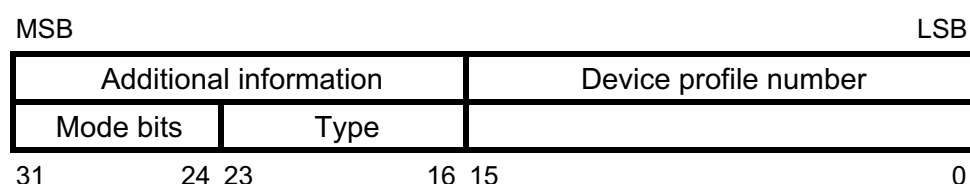
### 7.1 Predefined objects

The default values for communication objects 1000<sub>h</sub> to 1FFF<sub>h</sub> which are not defined by the communication profile (see /3/) are mentioned below.

#### 7.1.1 Object 1000<sub>h</sub>: Device type

The object at index 1000<sub>h</sub> describes the type of a device and its functionality.

For multi device modules the additional information parameter contains 0FFF<sub>h</sub> and the device profile number referenced by object 1000<sub>h</sub> is the device profile of the first device in the object dictionary. All other devices of a multiple device module identify their profiles at object 67FF<sub>h</sub> + x \* 800<sub>h</sub> with x = internal number of the device (0..7).



For devices in this device profile the following assignment exists:

Device	Additional information																Device profile number
	Mode bits								Type								
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 - 0
Frequency converter	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	1	0192 <sub>h</sub> = 402
Servo drive	*	*	*	*	*	*	*	*	0	0	0	0	0	0	1	0	0192 <sub>h</sub>
Stepper motor	*	*	*	*	*	*	*	*	0	0	0	0	0	1	0	0	0192 <sub>h</sub>
Multiple device module	*	*	*	*	1	1	1	1	1	1	1	1	1	1	1	1	0192 <sub>h</sub>

Table 3: Structure of the device type entry in the object dictionary  
(\* ... manufacturer-specific)

#### 7.1.2 Object 1001<sub>h</sub>: Error register

All bits are defined as in /3/. The device specific bit in the *error register* is used by the CANopen Device Profile for Drives and Motion Control. The error code can be read from the *predefined error field* at object 1003<sub>h</sub> and to be compatible with device profiles for drives available for other field bus systems from object 603F<sub>h</sub> as well.

#### 7.1.3 Object 67FF<sub>h</sub>: Single device type

The object at index 67FF<sub>h</sub> and multiples with an offset of 800<sub>h</sub> describe the type of each device within one drive unit and its functionality. The object structure is the same as defined in object 1000<sub>h</sub>.

### 7.2 PDO mapping

A drive supporting more than one mode will mostly use more than one standard PDO. Therefore a lot of PDOs are predefined in respect to the different possible modes of operation for drives.



The hereafter described PDO distribution should be used for every axle of a multi-device module with an offset of 64, e.g. the first PDO of the second axle gets the number 65. In this way a system with a maximum of 8 axles is supported.

It is open to a manufacturer to specify additional entries in the mapping table or define absolutely new PDO mappings and it is also open to a user to change these default settings by changing the mapping structure, if the module supports variable mapping on these PDOs.

### 7.2.1 Receive PDOs

PDO no.	Mapping object index	Mapping object name	M/O	Comment
1	6040 <sub>h</sub>	<i>Controlword</i>	M	controls the state machine
2	6040 <sub>h</sub> 6060 <sub>h</sub>	<i>Controlword</i> <i>Modes of operation</i>	O	controls the state machine and modes of operation
3	6040 <sub>h</sub> 607A <sub>h</sub>	<i>Controlword</i> <i>Target position</i>	O	controls the state machine and the target position (pp)
4	6040 <sub>h</sub> 60FF <sub>h</sub>	<i>Controlword</i> <i>Target velocity (pv)</i>	O	controls the state machine and the target velocity (pv)
5	6040 <sub>h</sub> 6071 <sub>h</sub>	<i>Controlword</i> <i>Target torque</i>	O	controls the state machine and the target torque (tq)
6	6040 <sub>h</sub> 6042 <sub>h</sub>	<i>Controlword</i> <i>Target velocity (vl)</i>	O	controls the state machine and the nominal speed (vl)
7	6040 <sub>h</sub> 60FE <sub>h</sub>	<i>Controlword</i> <i>Digital outputs</i>	O	controls the state machine and the digital outputs
8	6040 <sub>h</sub> 6060 <sub>h</sub>	<i>Controlword</i> <i>Modes of operation</i>	O	controls the state machine and mode of operation (Broadcast PDO)
9-20				reserved
21-64			O	manufacturer specific

#### 7.2.1.1 1<sup>st</sup> Receive PDO

Index	Sub	Name	Default value
1400 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1600 <sub>h</sub>	0	Number of mapped objects	1
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>

**7.2.1.2 2<sup>nd</sup> Receive PDO**

Index	Sub	Name	Default value
1401 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1601 <sub>h</sub>	0	Number of mapped objects	3
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Modes of operation</i>	6060 0008 <sub>h</sub>

**7.2.1.3 3<sup>rd</sup> Receive PDO**

Index	Sub	Name	Default value
1402 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1602 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Target position</i>	607A 0020 <sub>h</sub>

**7.2.1.4 4<sup>th</sup> Receive PDO**

Index	Sub	Name	Default value
1403 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1603 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Target velocity (pv)</i>	60FF 0020 <sub>h</sub>

**7.2.1.5 5<sup>th</sup> Receive PDO**

Index	Sub	Name	Default value
1404 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1604 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Target torque</i>	6071 0010 <sub>h</sub>

**7.2.1.6 6<sup>th</sup> Receive PDO**

Index	Sub	Name	Default value
1405 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1605 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Target velocity (vI)</i>	6042 0010 <sub>h</sub>

**7.2.1.7 7<sup>th</sup> Receive PDO**

Index	Sub	Name	Default value
1406 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1606 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Digital outputs</i>	60FE 0120 <sub>h</sub>

**7.2.1.8 8<sup>th</sup> Receive PDO**

Index	Sub	Name	Default value
1407 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1607 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Controlword</i>	6040 0010 <sub>h</sub>
	2	<i>Modes of operation</i>	6060 0008 <sub>h</sub>

**7.2.2 Transmit PDOs**

The task of the transmit PDOs is the monitoring of the drives behavior. The TPDO 1,2 and 7 are event driven. The other PDOs are synchronous.

PDO no.	Mapping object index	Mapping object name	M/O	Comment
1	6041 <sub>h</sub>	<i>Statusword</i>	M	shows status
2	6041 <sub>h</sub> 6061 <sub>h</sub>	<i>Statusword</i> <i>Modes of operation display</i>	O	shows status and the current mode of operation
3	6041 <sub>h</sub> 6064 <sub>h</sub>	<i>Statusword</i> <i>Position actual value</i>	O	shows status and the current position (pp)
4	6041 <sub>h</sub> 606C <sub>h</sub>	<i>Statusword</i> <i>Velocity actual value</i>	O	shows status and the current velocity (pv)
5	6041 <sub>h</sub> 6077 <sub>h</sub>	<i>Statusword</i> <i>Torque actual value</i>	O	shows status and the current torque (tq)
6	6041 <sub>h</sub> 6044 <sub>h</sub>	<i>Statusword</i> <i>vl control effort</i>	O	shows status and the current speed (vl)
7	6041 <sub>h</sub> 60FD <sub>h</sub>	<i>Statusword</i> <i>Digital inputs</i>	O	shows status and the digital inputs
8-20				reserved
21-64			O	manufacturer specific

**7.2.2.1 1<sup>st</sup> Transmit PDO**

Index	Sub	Name	Default value
1800 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A00 <sub>h</sub>	0	Number of mapped objects	1
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>

If the *statusword* changes its value, this PDO shall be transmitted immediately.

**7.2.2.2 2<sup>nd</sup> Transmit PDO**

Index	Sub	Name	Default value
1801 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A01 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>Modes of operation display</i>	6061 0008 <sub>h</sub>

**7.2.2.3 3<sup>rd</sup> Transmit PDO**

Index	Sub	Name	Default value
1802 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A02 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>Position actual value</i>	6064 0020 <sub>h</sub>

**7.2.2.4 4<sup>th</sup> Transmit PDO**

Index	Sub	Name	Default value
1803 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A03 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>Velocity actual value</i>	606C 0020 <sub>h</sub>

**7.2.2.5 5<sup>th</sup> Transmit PDO**

Index	Sub	Name	Default value
1804 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A04 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>Torque actual value</i>	6077 0010 <sub>h</sub>

**7.2.2.6 6<sup>th</sup> Transmit PDO**

Index	Sub-Index	Comment	Default Value
1805 <sub>h</sub>	0	number of entries	5
	1	COB-ID used by PDO	see /3/
	2	transmission type	255
	3	inhibit time	see /3/
	4	reserved	see /3/
	5	event timer	see /3/

Index	Sub	Name	Default value
1A05 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>VI control effort</i>	6044 0010 <sub>h</sub>

**7.2.2.7 7<sup>th</sup> Transmit PDO**

Index	Sub	Name	Default value
1806 <sub>h</sub>	0	Number of entries	5
	1	COB-ID used by PDO	see /3/
	2	Transmission type	255
	3	Inhibit time	see /3/
	4	Reserved	see /3/
	5	Event timer	see /3/

Index	Sub	Name	Default value
1A06 <sub>h</sub>	0	Number of mapped objects	2
	1	<i>Statusword</i>	6041 0010 <sub>h</sub>
	2	<i>Digital inputs</i>	60FD 0020 <sub>h</sub>

## 8 OBJECT DICTIONARY

Each drive shares the dictionary entries from 6000<sub>h</sub> to 63FF<sub>h</sub>. These entries are common to all drive modules and each module implements only the dictionary parts which are relevant for its functions.

Drives having also digital or analog I/O are using dictionary entries from 8000<sub>h</sub> to 83FF<sub>h</sub> as described in /6/ for the objects from 6000<sub>h</sub> to 63FF<sub>h</sub> with an offset of 2000<sub>h</sub>.



## 9 COMMON ENTRIES

### 9.1 General information

#### 9.1.1 Motor data

The objects 6402<sub>h</sub> to 64FF<sub>h</sub> serve as a database for motor parameters. The values are typically found on the motor's nameplate or the manufacturer's motor catalog and are used to maintain a service database within the controlling device of the drive. Most of the entries are typically entities from the manufacturer's motor catalog. Future drives should at least contain an entry to the electronically available catalog via a common net address, like a HTTP link to the manufacturers database, *http motor catalog address*.

The objects 6402<sub>h</sub> to 640F<sub>h</sub> are highly recommended.

Some objects are available in the object dictionary of other field bus systems, so their indices are not in the default range from 6400<sub>h</sub> to 64FF<sub>h</sub>.

There is one manufacturer specific data RECORD at object 6410<sub>h</sub>. It should contain as much as possible entries for the used motor. The structure of this record is described in the manufacturer's data sheet for the drive unit.

#### 9.1.2 Drive data

The objects 6500<sub>h</sub> to 65FF<sub>h</sub> serve as a database for drive parameters.

There is one manufacturer specific data RECORD at object 6510<sub>h</sub>. It should contain as much as possible entries for the used drive. The structure of this record is described in the manufacturer's handbook. The data must be filled in while in commissioning. The values are typically found on the drive's datasheet or the manufacturer's drives catalog and are used to maintain a service database within the controlling device of the drive.

Most of the entries are typically entities from the manufacturer's drive catalog. Future drives should at least contain an entry to the electronically available catalog via a common net address, like a HTTP link to the manufacturers database, *http drive catalog address*.

In /3/ three optional objects for a CANopen device are recommended:

Index	Name
1008 <sub>h</sub>	<i>Manufacturer device name</i>
1009 <sub>h</sub>	<i>Manufacturer hardware version</i>
100A <sub>h</sub>	<i>Manufacturer software version</i>

## 9.2 Object dictionary entries

### 9.2.1 Objects defined in this chapter

Index	Object	Name	Type	Attr.	M/O
6007 <sub>h</sub>	VAR	<i>Abort connection option code</i>	INTEGER16	rw	O
603F <sub>h</sub>	VAR	<i>Error code</i>	UNSIGNED16	ro	O
6402 <sub>h</sub>	VAR	<i>Motor type</i>	UNSIGNED16	rw	O
6403 <sub>h</sub>	VAR	<i>Motor catalog number</i>	VISIBLE_STRING	rw	O
6404 <sub>h</sub>	VAR	<i>Motor manufacturer</i>	VISIBLE_STRING	rw	O
6405 <sub>h</sub>	VAR	<i>http motor catalog address</i>	VISIBLE_STRING	rw	O
6406 <sub>h</sub>	VAR	<i>Motor calibration date</i>	TIME_OF_DAY	rw	O
6407 <sub>h</sub>	VAR	<i>Motor service period</i>	UNSIGNED32	rw	O
6410 <sub>h</sub>	RECORD	<i>Motor data</i>	(manufacturer specific)	rw	O
6502 <sub>h</sub>	VAR	<i>Supported drive modes</i>	UNSIGNED32	ro	O
6503 <sub>h</sub>	VAR	<i>Drive catalog number</i>	VISIBLE_STRING	ro	O
6504 <sub>h</sub>	VAR	<i>Drive manufacturer</i>	VISIBLE_STRING	ro	O
6505 <sub>h</sub>	VAR	<i>http drive catalog address</i>	VISIBLE_STRING	rw	O
6510 <sub>h</sub>	RECORD	<i>Drive data</i>	(manufacturer specific)	rw	O
60FD <sub>h</sub>	VAR	<i>Digital inputs</i>	UNSIGNED32	rw	O
60FE <sub>h</sub>	ARRAY	<i>Digital outputs</i>	UNSIGNED32	rw	O

## 9.3 Object description

The drive functionality in error cases is adjustable by the following objects.

### 9.3.1 Object 6007<sub>h</sub>: *Abort connection option code*

The content of this object selects the function to be performed when the connection to the network is lost.

#### OBJECT DESCRIPTION

INDEX	6007 <sub>h</sub>
Name	Abort connection option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

## DATA DESCRIPTION

Option code	Meaning
0	no action
1	malfunction
2	Device control command 'Disable Voltage'
3	Device control command 'Quick Stop'
4..32767	reserved
-32768..-1	manufacturer specific

**9.3.2 Object 603F<sub>h</sub>: Error code**

The *Error code* captures the code of the last error that occurred in the drive. It corresponds to the value of the lower 16 bits of object 1003<sub>h</sub> *pre-defined error field*.

## OBJECT DESCRIPTION

INDEX	603F <sub>h</sub>
Name	Error code
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	0

**9.3.3 Object 6402<sub>h</sub>: Motor type**

The type of motor driven by the controller.

## OBJECT DESCRIPTION

INDEX	6402 <sub>h</sub>
Name	Motor type
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

## DATA DESCRIPTION

Value	Motor Type
0000 <sub>h</sub>	Non-standard motor
0001 <sub>h</sub>	Phase modulated DC motor
0002 <sub>h</sub>	Frequency controlled DC motor
0003 <sub>h</sub>	PM synchronous motor
0004 <sub>h</sub>	FC synchronous motor
0005 <sub>h</sub>	Switched reluctance motor
0006 <sub>h</sub>	Wound rotor induction motor
0007 <sub>h</sub>	Squirrel cage induction motor
0008 <sub>h</sub>	Stepper motor
0009 <sub>h</sub>	Micro-step stepper motor
000A <sub>h</sub>	Sinusoidal PM BL motor
000B <sub>h</sub>	Trapezoidal PM BL motor
000C <sub>h</sub>	reserved
.....	.....
7FFF <sub>h</sub>	reserved
8000 <sub>h</sub>	manufacturer specific
.....	.....
FFFF <sub>h</sub>	manufacturer specific

#### 9.3.4 Object 6403<sub>h</sub>: *Motor catalog number*

The manufacturer's motor catalog number (nameplate number).

## OBJECT DESCRIPTION

INDEX	6403 <sub>h</sub>
Name	Motor catalog number
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.5 Object 6404<sub>h</sub>: Motor manufacturer**

The motor manufacturer's name.

**OBJECT DESCRIPTION**

INDEX	6404 <sub>h</sub>
Name	Motor manufacturer
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.6 Object 6405<sub>h</sub>: http motor catalog address****OBJECT DESCRIPTION**

INDEX	6405 <sub>h</sub>
Name	http motor catalog address
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.7 Object 6406<sub>h</sub>: Motor calibration date**

Date of the last motor inspection.

**OBJECT DESCRIPTION**

INDEX	6406 <sub>h</sub>
Name	Motor calibration date
Object Code	VAR
Data Type	TIME_OF_DAY
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

**9.3.8 Object 6407<sub>h</sub>: *Motor service period***

Value in hours of the nominal motor lifetime. The motor needs service after this time.

**OBJECT DESCRIPTION**

INDEX	6407 <sub>h</sub>
Name	Motor service period
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**9.3.9 Object 6410<sub>h</sub>: *Motor data***

This object contains as much as possible information about the connected motor. The structure of this record is described in the drive manufacturer's handbook.

**OBJECT DESCRIPTION**

INDEX	6410 <sub>h</sub>
Name	Motor data
Object Code	RECORD
Data Type	manufacturer specific
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	Number of entries
Entry category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 ... 254
Default Value	No

Sub-Index	1
Description	manufacturer specific
Entry category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

Sub-Index	2
Description	manufacturer specific
Entry category	Optional
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry category	Optional
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

### 9.3.10 Object 6502<sub>h</sub>: *Supported drive modes*

A drive can support more than one and several distinct modes of operation. This object gives an overview of the implemented operating modes in the device. This object is read only.

#### OBJECT DESCRIPTION

INDEX	6502 <sub>h</sub>
Name	Supported drive modes
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

#### ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

#### DATA DESCRIPTION

31	16	15	7	6	5	4	3	2	1	0		
manufacturer specific				reserved		ip	hm	reserved	tq	pv	vl	pp
MSB				LSB								

**9.3.11 Object 6503<sub>h</sub>: Drive catalog number**

The manufacturer's drive catalog number (nameplate number).

**OBJECT DESCRIPTION**

INDEX	6503 <sub>h</sub>
Name	Drive catalog number
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.12 Object 6504<sub>h</sub>: Drive manufacturer**

The drive manufacturer's name.

**OBJECT DESCRIPTION**

INDEX	6504 <sub>h</sub>
Name	Drive manufacturer
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.13 Object 6505<sub>h</sub>: http drive catalog address**

The internet address of the manufacturer.

**OBJECT DESCRIPTION**

INDEX	6505 <sub>h</sub>
Name	http drive catalog address
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional



## ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	No
Default Value	No

**9.3.14 Object 6510<sub>h</sub>: Drive data**

This object contains as much as possible information about the drive unit. The structure of this record is described in the drive manufacturer's handbook.

## OBJECT DESCRIPTION

Index	6510 <sub>h</sub>
Name	Drive data
Object Code	RECORD
Data Type	manufacturer specific
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 ... 254
Default Value	No

Sub-Index	1
Description	manufacturer specific
Entry category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

Sub-Index	2
Description	manufacturer specific
Entry category	Optional
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry category	Optional
Access	rw
PDO Mapping	Possible
Value Range	No
Default Value	No

### 9.3.15 Object 60FD<sub>h</sub>: *Digital inputs*

This index defines simple digital inputs for drives. The user may apply any signals to these inputs for special purposes like limit or reference switches.

#### OBJECT DESCRIPTION

INDEX	60FD <sub>h</sub>
Name	Digital inputs
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

#### ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

#### DATA DESCRIPTION

31	16	15	4	3	2	1	0
manufacturer specific	reserved		interlock	home switch	positive limit switch	negative limit switch	

MSB

LSB

The switch have to be "active high".

### 9.3.16 Object 60FE<sub>h</sub>: *Digital outputs*

This index defines simple digital outputs for drives.

#### OBJECT DESCRIPTION

Index	60FE <sub>h</sub>
Name	Digital outputs
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

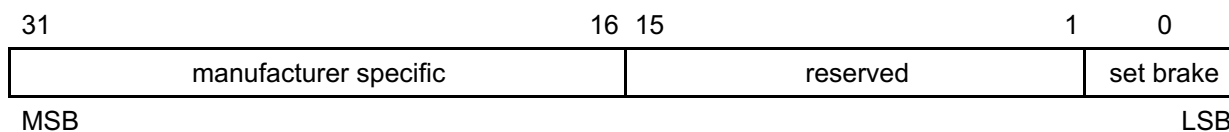
Sub-Index	0
Description	number of entries
Entry description	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 ... 2
Default Value	No

Sub-Index	1
Description	Physical outputs
Entry description	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

Sub-Index	2
Description	Bit mask
Entry description	Optional
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

## DATA DESCRIPTION

The first sub-index defines the assigned outputs.



The second sub-index describes a mask to specify which of the outputs shall be used:

- |   |   |                    |
|---|---|--------------------|
| 0 | - | Output is disabled |
| 1 | - | Output is enabled  |

# 10 DEVICE CONTROL

## 10.1 General information

The device control function block controls all functions of the drive (drive function and power section). It is divided into:

- device control of the state machine
- operation mode function

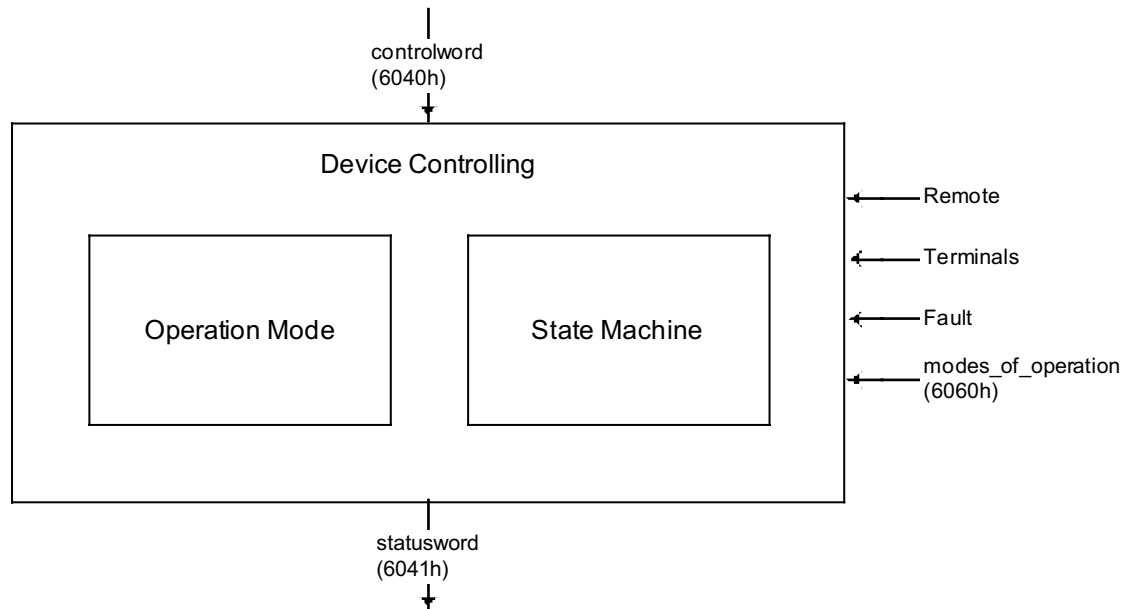


Figure 5: Device controlling

The state of the drive can be controlled by the *controlword*.

The state of the drive is shown in the *statusword*.

In remote mode the device is controlled directly from the CANopen network by PDO and SDO.

The state machine is controlled externally by the *controlword* and external signals. The write access to the *controlword* is controlled by the optional hardware signal 'Remote'. The state machine is also controlled by internal signals like faults and *modes of operation*.

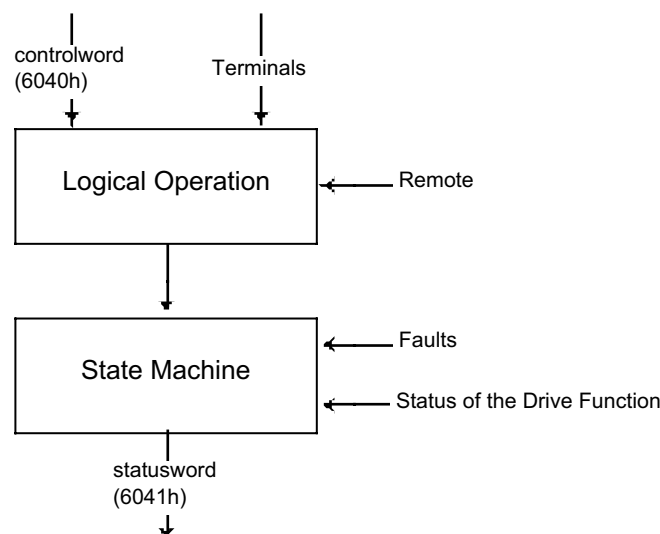


Figure 6: Remote mode

### 10.1.1 State machine

The state machine describes the device status and the possible control sequence of the drive. A single state represents a special internal or external behavior. The state of the drive also determines which commands are accepted. E.g. it is only possible to start a point-to-point move when the drive is in state OPERATION ENABLE.

States may be changed using the *controlword* and/or according to internal events. The current state can be read using the *statusword*.

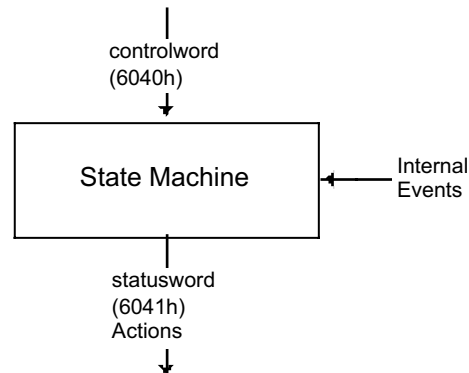


Figure 7: State machine in system context

The state machine in Figure 7 describes the state machine of the device with respect to control of the power electronics as a result of user commands and internal drive faults.

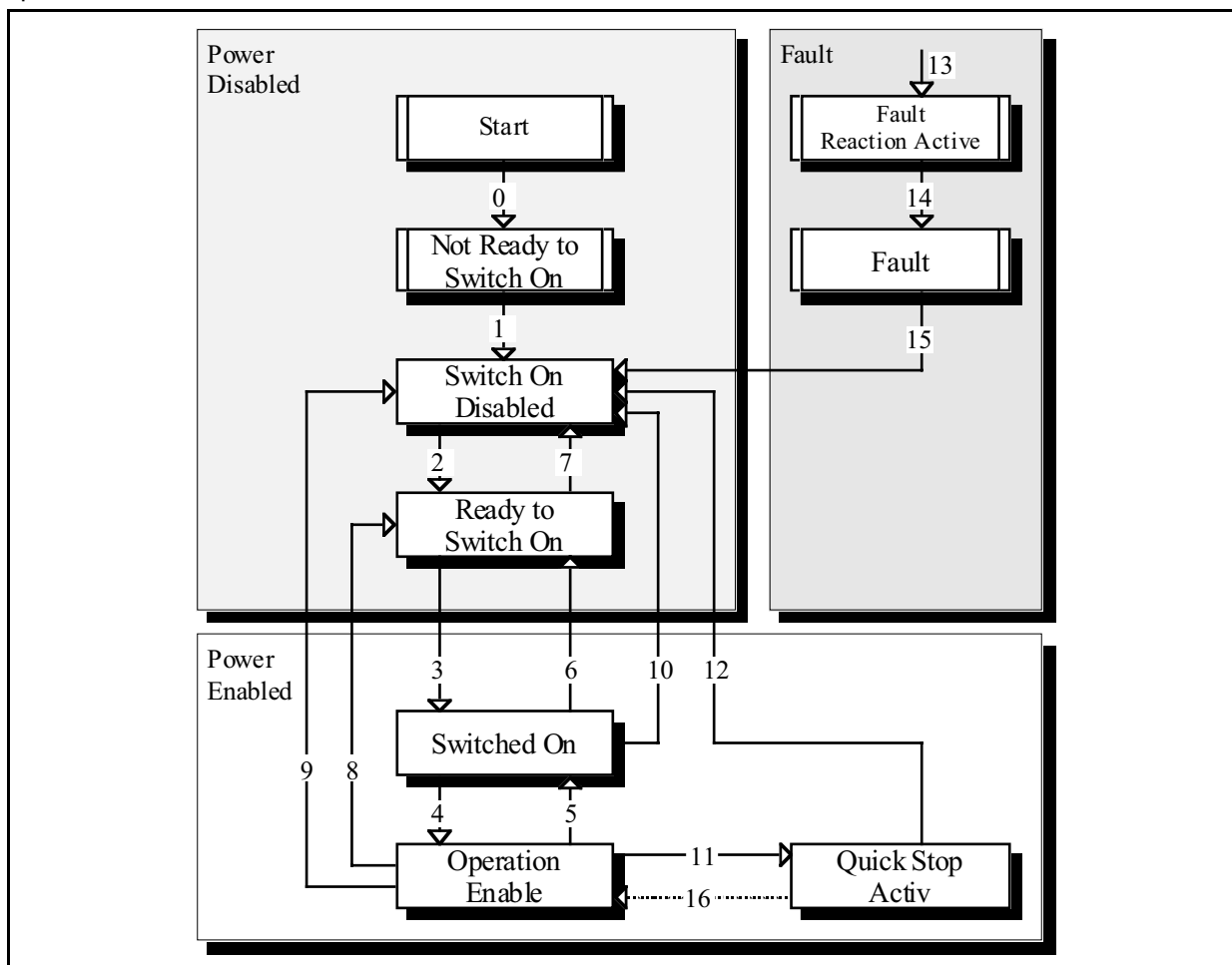


Figure 8: State machine

### 10.1.1.1 Drive states

The drive states may become more evident when considering the following (generic) block diagram of a drive:

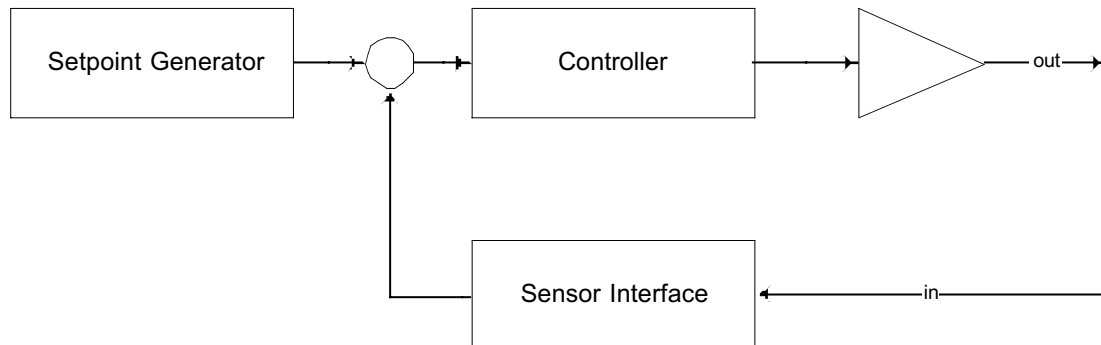


Figure 9: Generic control loop block diagram

The sensor interface and the “in”-terminal are only present in drives with a feedback path. Normally the set-point generator, the controller and the power amplifier can be disabled.

The following states of the device are possible:

- **NOT READY TO SWITCH ON:**  
*Low level power (e.g.  $\pm 15V$ , 5V) has been applied to the drive.  
 The drive is being initialized or is running self test.  
 A brake, if present, has to be applied in this state.  
 The drive function is disabled.*
- **SWITCH ON DISABLED:**  
*Drive initialization is complete.  
 The drive parameters have been set up.  
 Drive parameters may be changed.  
 High voltage may not be applied to the drive, (e.g. for safety reasons).  
 The drive function is disabled.*
- **READY TO SWITCH ON:**  
*High voltage may be applied to the drive.  
 The drive parameters may be changed.  
 The drive function is disabled.*
- **SWITCHED ON:**  
*High voltage has been applied to the drive.  
 The power amplifier is ready.  
 The drive parameters may be changed.  
 The drive function is disabled.*
- **OPERATION ENABLE:**  
*No faults have been detected.  
 The drive function is enabled and power is applied to the motor.  
 The drive parameters may be changed.  
 (This corresponds to normal operation of the drive.)*
- **QUICK STOP ACTIVE:**  
*The drive parameters may be changed.  
 The quick stop function is being executed.  
 The drive function is enabled and power is applied to the motor.*
  - If the *quick stop option code* is switched to 5 (stay in the state QUICK STOP ACTIVE), you can't leave the state QUICK STOP ACTIVE, but you can transmit to the state OPERATION ENABLE with the command 'Enable Operation'.

- **FAULT REACTION ACTIVE:**  
*The drive parameters may be changed.*  
*A fault has occurred in the drive.*  
*The quick stop function is being executed.*  
*The drive function is enabled and power is applied to the motor.*
- **FAULT:**  
*The drive parameters may be changed.*  
*A fault has occurred in the drive.*  
*High voltage switch-on/-off depends on the application.*  
*The drive function is disabled.*

#### 10.1.1.2 State transitions of the drive supervisor

State transitions are caused by internal events in the drive or by commands from the host via the *controlword*.

- **State Transition 0: START ⇒ NOT READY TO SWITCH ON**  
*Event: Reset.*  
*Action: The drive self-tests and/or self-initializes.*
- **State Transition 1: NOT READY TO SWITCH ON ⇒ SWITCH ON DISABLED**  
*Event: The drive has self-tested and/or initialized successfully.*  
*Action: Activate communication.*
- **State Transition 2: SWITCH ON DISABLED ⇒ READY TO SWITCH ON**  
*Event: 'Shutdown' command received from host.*  
*Action: None*
- **State Transition 3: READY TO SWITCH ON ⇒ SWITCHED ON**  
*Event: 'Switch On' command received from host.*  
*Action: The power section is switched on if it is not already switched on.*
- **State Transition 4: SWITCHED ON ⇒ OPERATION ENABLE**  
*Event: 'Enable Operation' command received from host.*  
*Action: The drive function is enabled.*
- **State Transition 5: OPERATION ENABLE ⇒ SWITCHED ON**  
*Event: 'Disable Operation' command received from host.*  
*Action: The drive operation will be disabled.*
- **State Transition 6: SWITCHED ON ⇒ READY TO SWITCH ON**  
*Event: 'Shutdown' command received from host.*  
*Action: The power section is switched off.*
- **State Transition 7: READY TO SWITCH ON ⇒ SWITCH ON DISABLED**  
*Event: 'Quick Stop' and 'Disable Voltage' command received from host.*  
*Action: None*
- **State Transition 8: OPERATION ENABLE ⇒ READY TO SWITCH ON**  
*Event: 'Shutdown' command received from host.*  
*Action: The power section is switched off immediately, and the motor is free to rotate if unbraked.*
- **State Transition 9: OPERATION ENABLE ⇒ SWITCH ON DISABLED**  
*Event: 'Disable Voltage' command received from host.*  
*Action: The power section is switched off immediately, and the motor is free to rotate if unbraked.*
- **State Transition 10: SWITCHED ON ⇒ SWITCH ON DISABLED**  
*Event: 'Disable Voltage' or 'Quick Stop' command received from host.*  
*Action: The power section is switched off immediately, and the motor is free to rotate if unbraked.*

- State Transition 11: OPERATION ENABLE  $\Rightarrow$  QUICK STOP ACTIVE  
Event: 'Quick Stop' command received from host.  
Action: The quick stop function is executed.
- State Transition 12: QUICK STOP ACTIVE  $\Rightarrow$  SWITCH ON DISABLED  
Event: 'Quick Stop' is completed or 'Disable Voltage' command received from host.  
This transition is possible, if the Quick-Stop-Option-Code is different 5 (stay in the state 'Quick Stop Active').  
Action: The power section is switched off.
- State Transition 13: All states  $\Rightarrow$  FAULT REACTION ACTIVE  
A fault has occurred in the drive.  
Action: Execute appropriate fault reaction.
- State Transition 14: FAULT REACTION ACTIVE  $\Rightarrow$  FAULT  
Event: The fault reaction is completed.  
Action: The drive function is disabled. The power section may be switched off.
- State Transition 15: FAULT  $\Rightarrow$  SWITCH ON DISABLED  
Event: 'Fault Reset' command received from host.  
Action: A reset of the fault condition is carried out if no fault exists currently on the drive.  
After leaving the state Fault the Bit 'Fault Reset' of the controlword has to be cleared by the host.
- State Transition 16: QUICK STOP ACTIVE  $\Rightarrow$  OPERATION ENABLE  
Event: 'Enable Operation' command received from host. This transition is possible if the Quick-Stop-Option-Code is 5, 6, 7 or 8 ( $\rightarrow$  Chapter 10.3.5).  
Action: The drive function is enabled.

**Notes:**

If a command is received which causes a change of state, this command must be processed completely and the new state attained before the next command can be processed.

'Drive function is disabled' implies no energy is supplied to the motor. This may be achieved by different manufacturers in different ways. Reference values are not processed.

'Drive function is enabled' implies that energy can be supplied to the motor. The reference values (torque, velocity, position) are processed.

'Fault occurred' implies that a fault in the drive has occurred. In this case there is a transition to the state FAULT REACTION ACTIVE. In this state the device will execute a special fault reaction. After the execution of this fault reaction the device will switch to the state FAULT. This state can only be left by the command 'Fault Reset', but only if the fault is not active any more.

**10.2 Object dictionary entries****10.2.1 Objects defined in this chapter**

Index	Object	Name	Type	Attr.	M/O
6040 <sub>h</sub>	VAR	Controlword	UNSIGNED16	rw	M
6041 <sub>h</sub>	VAR	Statusword	UNSIGNED16	ro	M
605B <sub>h</sub>	VAR	Shutdown option code	INTEGER16	rw	O
605C <sub>h</sub>	VAR	Disable operation option code	INTEGER16	rw	O
605A <sub>h</sub>	VAR	Quick stop option code	INTEGER16	rw	O
605D <sub>h</sub>	VAR	Halt option code	INTEGER16	rw	O
605E <sub>h</sub>	VAR	Fault reaction option code	INTEGER16	rw	O
6060 <sub>h</sub>	VAR	Modes of operation	INTEGER8	rw	M
6061 <sub>h</sub>	VAR	Modes of operation display	INTEGER8	ro	M



### 10.3 Object description

#### 10.3.1 Object 6040<sub>h</sub>: *Controlword*

The *controlword* consist of bits for:

- the controlling of the state,
- the controlling of operating modes and
- manufacturer specific options.

#### OBJECT DESCRIPTION

INDEX	6040 <sub>h</sub>
Name	Controlword
Object Code	VAR
Data Type	UNSIGNED16
Category	Mandatory

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

#### DATA DESCRIPTION

The bits of the *controlword* are defined as follows:

15	11	10	9	8	7	6	4	3	2	1	0
manufacturer specific	reserved	halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch on			
O	O	O	M	O	M	M	M	M			
MSB				LSB							
0	-	Optional			M	-	Mandatory				

#### **BITS 0 – 3 AND 7:**

Device control commands are triggered by the following bit patterns in the *controlword*:


Command	Bit of the <i>controlword</i>					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset		X	X	X	X	15

Table 4: Device control commands (bits marked X are irrelevant, \* ... In the state SWITCHED ON the drive executes the functionality of this state., \*\* ... It exists no functionality in the state SWITCHED ON. The drive does not do any in this state.)

**BITS 4, 5, 6 AND 8:**

These bits are operation mode specific. The description is situated in the chapter of the special mode. The following table gives an overview:

Bit	Operation mode					
	Velocity mode	Profile position mode	Profile velocity mode	Profile torque mode	Homing mode	Interpolation position mode
4	rfg enable	New set-point	reserved	reserved	Homing operation start	Enable ip mode
5	rfg unlock	Change set immediately	reserved	reserved	reserved	reserved
6	rfg use ref	abs / rel	reserved	reserved	reserved	reserved
8	Halt	Halt	Halt	Halt	Halt	Halt

Table 5: Mode specific bits in the *controlword***BITS 9, 10:**

These bits are reserved for further use. They are inactive by setting to zero. If they have no special function, they must be set to zero.

**BITS 11, 12, 13, 14 AND 15:**

These bits are manufacturer specific.

**10.3.2 Object 6041<sub>h</sub>: Statusword**

The *statusword* indicates the current state of the drive. No bits are latched. The *statusword* consist of bits for:

- the current state of the drive,
- the operating state of the mode and
- manufacturer specific options.

**OBJECT DESCRIPTION**

INDEX	6041 <sub>h</sub>
Name	Statusword
Object Code	VAR
Data Type	UNSIGNED16
Category	Mandatory

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**DATA DESCRIPTION**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

MSB

LSB

Bit	Description	M / O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12 - 13	Operation mode specific	O
14 - 15	Manufacturer specific	O

Table 6: Bits in the *statusword***BITS 0 – 3, 5 AND 6:**

The following bits indicate the status of the device:

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Table 7: Device state bits (x ... irrelevant for this state)

**BIT 4: VOLTAGE ENABLED**

High voltage is applied to the drive when this bit is set to 1.

**BIT 5: QUICK STOP**

When reset, this bit indicates that the drive is reacting on a quick stop request. Bits 0, 1 and 2 of the *statusword* must be set to 1 to indicate that the drive is capable to regenerate. The setting of the other bits indicates the status of the drive (e.g. the drive is performing a quick stop as result of a reaction to a non-fatal fault. The fault bit is set as well as bits 0, 1 and 2).

**BIT 7: WARNING**

A drive warning is present if bit 7 is set. The cause means no error but a state that has to be mentioned, e.g. temperature limit, job refused. The status of the drive does not change. The cause of this warning may be found by reading the fault code parameter. The bit is set and reset by the device.

**BIT 8:**

This bit may be used by a drive manufacturer to implement any manufacturer specific functionality.

**BIT 9: REMOTE**

If bit 9 is set, then parameters may be modified via the CAN-network, and the drive executes the content of a command message. If the bit remote is reset, then the drive is in local mode and will not execute the command message. The drive may transmit messages containing valid actual values like a *position actual value*, depending on the actual drive configuration. The drive will accept accesses via SDO in local mode.

**BIT 10: TARGET REACHED**

If bit 10 is set by the drive, then a set-point has been reached. The set-point is dependent on the operating mode. The description is situated in the chapter of the special mode. The change of a target value by software alters this bit.

If *quick stop option code* is 5, 6, 7 or 8, this bit must be set, when the quick stop operation is finished and the drive is halted.

If halt occurred and the drive has halted then this bit is set too.

**BIT 11: INTERNAL LIMIT ACTIVE**

This bit set by the drive indicates, that an internal limitation is active (e.g. *position range limit*).

**BIT 12 AND 13:**

These bits are operation mode specific. The description is situated in the chapter of the special mode. The following table gives an overview:

Bit	Operation mode					
	vl	pp	pv	tq	hm	ip
12	reserved	Set-point acknowledge	Speed	reserved	Homing attained	ip mode active
13	reserved	Following error	Max slippage error	reserved	Homing error	reserved

Table 8: Mode specific bits in the *statusword*

**BIT 14 AND 15:**

These bits may be used by a drive manufacturer to implement any manufacturer specific functionality.

**10.3.3 Object 605B<sub>n</sub>: Shutdown option code**

The parameter *shutdown option code* determines what action should be taken if there is a transition

OPERATION ENABLE ⇒ READY TO SWITCH ON.

**OBJECT DESCRIPTION**

INDEX	605B <sub>n</sub>
Name	Shutdown option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	No
Value Range	INTEGER16
Default Value	0

## DATA DESCRIPTION

Value	Description
-32768 ... -1	manufacturer specific
0	Disable drive function
1	Slow down with slow down ramp; disable of the drive function
2 ... 32767	reserved

**10.3.4 Object 605C<sub>h</sub>: Disable operation option code**

The parameter *disable operation option code* determines what action should be taken if there is a transition

OPERATION ENABLE ⇒ SWITCHED ON.

## OBJECT DESCRIPTION

INDEX	605C <sub>h</sub>
Name	Disable operation option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	INTEGER16
Default Value	1

## DATA DESCRIPTION

Value	Description
-32768 ... -1	manufacturer specific
0	Disable drive function
1	Slow down with slow down ramp and then disabling of the drive function
2 ... 32767	reserved

**10.3.5 Object 605A<sub>h</sub>: Quick stop option code**

The parameter *quick stop option code* determines what action should be taken if the Quick Stop Function is executed.

## OBJECT DESCRIPTION

INDEX	605A <sub>h</sub>
Name	Quick stop option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	INTEGER16
Default Value	2

## DATA DESCRIPTION

Value	Description
-32768 ... -1	manufacturer Specific
0	disable drive function
1	slow down on slow down ramp
2	slow down on quick stop ramp
3	slow down on the current limit
4	slow down on the voltage limit
5	slow down on slow down ramp and stay in QUICK STOP
6	slow down on quick stop ramp and stay in QUICK STOP
7	slow down on the current limit and stay in QUICK STOP
8	slow down on the voltage limit and stay in QUICK STOP
9 ... 32767	Reserved

**10.3.6 Object 605D<sub>n</sub>: Halt option code**

The parameter *halt option code* determines what action should be taken if the bit 8 (halt) in the *controlword* is active.

## OBJECT DESCRIPTION

INDEX	605D <sub>n</sub>
Name	Halt option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	INTEGER16
Default Value	1

## DATA DESCRIPTION

Value	Description
-32768 ... -1	manufacturer Specific
0	disable drive, motor is free to rotate
1	slow down on slow down ramp
2	slow down on quick stop ramp

Value	Description
3	slow down on the current limit
4	slow down on the voltage limit
5 ... 32767	reserved

### 10.3.7 Object 605E<sub>h</sub>: Fault reaction option code

The parameter *fault reaction option code* determines what action should be taken if a fault occurs in the drive.

#### OBJECT DESCRIPTION

INDEX	605E <sub>h</sub>
Name	Fault reaction option code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	INTEGER16
Default Value	2

#### DATA DESCRIPTION

Value	Description
-32768 ... -1	manufacturer specific
0	disable drive, motor is free to rotate
1	slow down on slow down ramp
2	slow down on quick stop ramp
3	slow down on the current limit
4	slow down on the voltage limit
5 ... 32767	reserved

### 10.3.8 Object 6060<sub>h</sub>: Modes of operation

The parameter *modes of operation* switches the actually choosen operation mode.

#### OBJECT DESCRIPTION

INDEX	6060 <sub>h</sub>
Name	Modes of operation
Object Code	VAR
Data Type	INTEGER8
Category	Mandatory

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	No

## DATA DESCRIPTION

Value	Description
-1 ... -128	manufacturer specific modes of operation
0	reserved
1	Profile Position Mode
2	Velocity Mode
3	Profile Velocity Mode
4	Torque Profile Mode
5	reserved
6	Homing Mode
7	Interpolated Position Mode
8 ... 127	reserved

## NOTE

A read of *modes of operation* shows only the value of *modes of operation*. The actual mode of the drive is reflected in the object *modes of operation display*. It may be changed by writing to *modes of operation*.

**10.3.9 Object 6061h: Modes of operation display**

The *modes of operation display* shows the current mode of operation. The meaning of the returned value corresponds to that of the *modes of operation* option code (index 6060<sub>h</sub>).

## OBJECT DESCRIPTION

INDEX	6061 <sub>h</sub>
Name	Modes of operation display
Object Code	VAR
Data Type	INTEGER8
Category	Mandatory

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	No

## DATA DESCRIPTION

Same as for object 6060<sub>h</sub> *modes of operation*.

## NOTE

The actual mode is reflected in the *modes of operation display* (index 6061<sub>h</sub>), and not in the *modes of operation* (index 6060<sub>h</sub>).



## 10.4 Functional description

### 10.4.1 Modes of operation function

The device behavior depends on the activated modes of operation.

It is possible to implement different device modes. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. An example of exclusive functions are those for position and torque control, which can only control one variable at any one time. The variables can perform at most a limited function. Such hybrids are regarded as the particular characteristics of a mode of operation. Position control operation and encoder profile support can be active at the same time, for example. Consequently encoder profile support is not regarded as a mode of operation.

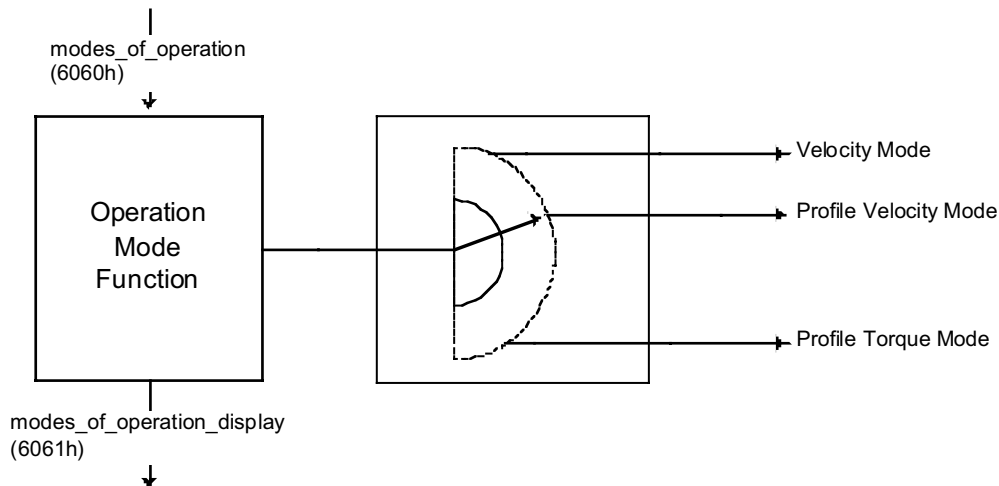


Figure 10: Operation mode function

It is possible for the user to switch between the various modes of operation as long as this is supported by the device. It is possible for the manufacturer to allow dynamic switching between different modes of operation at any time or to limit switching for example to the state SWITCHED ON. Switching can also be limited to the state 'local control'; i.e. not possible via the CAN-network. A device characteristic listed in the device function list can possibly have several modes of operation.

The following modes of operation are listed:

- *Velocity Mode (AC/DC drives, no feedback)*
- *Profile Velocity Mode (servo drives, feedback)*
- *Torque Profile Mode*
- *Homing Mode*
- *Profile Position Mode*
- *Interpolated Position Mode*

With the exception of the 'Homing Mode', these listed modes of operation can all be put under the heading of 'set-point setting'.

In parallel to this, manufacturer-specific modes of operation may also be available. These are not limited to set-point settings.

The reference operation is regarded as a special form of a program function. The program function allows the user to run complex of time-critical sequence, e. g. tool change or special reference operations, directly in the device.

The switching between the modes of operation listed above should not incur any automatic reconfiguration of the process data channel. Problems which occur through switching of set-point values during change of operating modes must be monitored by the user. If necessary they can be rectified by prior reconfiguration of the process data channel.

Two objects are defined for management of the modes of operation:

- *modes of operation*
- *modes of operation display*

The *statusword* contains bits, whose meaning is dependent on the mode of operation. When switching the mode of operation, the bits changing their meaning need to be monitored.

### 10.4.2 Drive disabling function

The drive disabling function defines the behavior of the drive when transitioning from the state OPERATION ENABLE to the state READY TO SWITCH ON ('Shutdown' command) or the state SWITCHED ON ('Disable Operation' command).

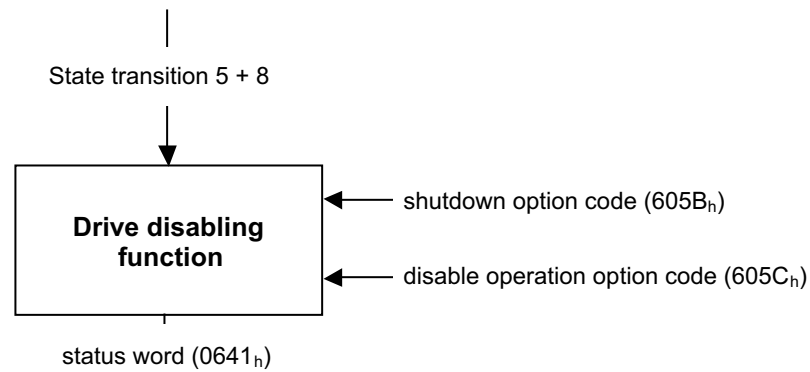


Figure 11: Modes of operation function

### 10.4.3 Quick stop function

The quick stop function is triggered by the 'Quick Stop' command.

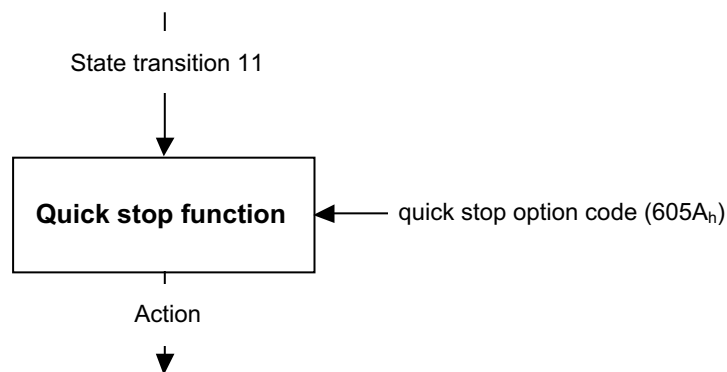


Figure 12: Quick stop function

### 10.4.4 Stop function

The stop function may be triggered by resetting the bit 'RFG-disable' in the *controlword*.

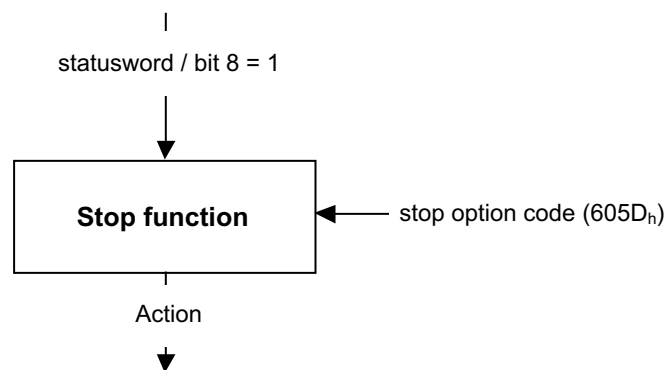


Figure 13: Stop function

### **10.4.5      Fault reaction**

Drive faults may be classified as fatal or non-fatal faults.

#### **10.4.5.1      Fatal faults**

When a fatal fault occurs, the drive is no longer able to control the motor, so an immediate switch-off of the drive is necessary.

#### **10.4.5.2      Non-fatal faults**

When a non-fatal fault occurs, the drive can run the motor in a controlled fashion. The actions which are executed depend on the *fault reaction option code*.

Once a fault occurs the drive will always enter the FAULT state, even if the fault clears before the drive enters the FAULT state. The FAULT state may only be left if the 'Fault Reset' command is received from a host, and no further fault is present in the drive.

# 11 FACTOR GROUP

## 11.1 General information

### 11.1.1 Factors

There is a need to interchange physical dimensions and sizes into the device internal units. To implement the interchange, several factors are necessary. This chapter describes how these factors have an influence on the system, how they are calculated and which data is necessary to build them. Normalized parameters are denoted with an asterisk \*.

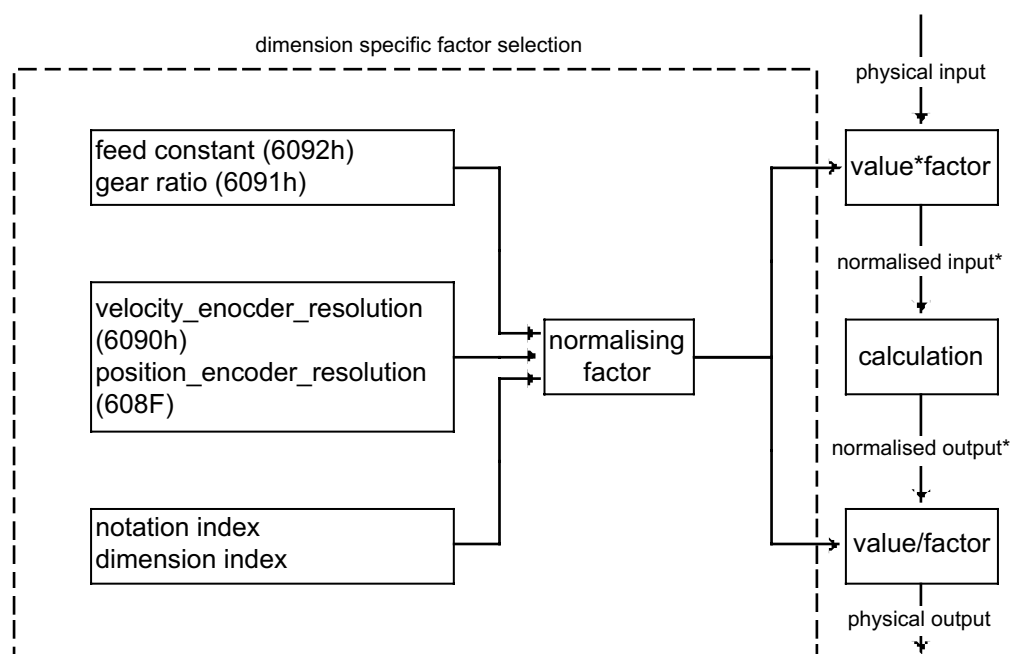


Figure 14: Influence of factors

### 11.1.2 Relationship between physical and internal units

The factors defined in the factor group set up a relationship between device internal units and physical units.

The factors are result of the calculation of two parameters called dimension index and notation index, which are defined in **Table D (see appendix Definition of dimension indices)**. One parameter indicates the physical dimension, the other the decimal exponent for the values. These factors are directly used to normalize the physical values.

The application specific parameters will be used in the corresponding mode of operation to build the described factors.

Parameters that are commonly used will be integrated in the object dictionary without defining their junctions. This guaranties a common parameter number for further use without the need for a predefinition.

## 11.2 Object dictionary entries

### 11.2.1 Objects defined in this chapter

Index	Object	Name	Type	Attr.	M/O
6089 <sub>h</sub>	VAR	<i>Position notation index</i>	INTEGER8	rw	O
608A <sub>h</sub>	VAR	<i>Position dimension index</i>	UNSIGNED8	rw	O
608B <sub>h</sub>	VAR	<i>Velocity notation index</i>	INTEGER8	rw	O
608C <sub>h</sub>	VAR	<i>Velocity dimension index</i>	UNSIGNED8	rw	O
608D <sub>h</sub>	VAR	<i>Acceleration notation index</i>	INTEGER8	rw	O
608E <sub>h</sub>	VAR	<i>Acceleration dimension index</i>	UNSIGNED8	rw	O
608F <sub>h</sub>	ARRAY	<i>Position encoder resolution</i>	UNSIGNED32	rw	O
6090 <sub>h</sub>	ARRAY	<i>Velocity encoder resolution</i>	UNSIGNED32	rw	O
6091 <sub>h</sub>	ARRAY	<i>Gear ratio</i>	UNSIGNED32	rw	O
6092 <sub>h</sub>	ARRAY	<i>Feed constant</i>	UNSIGNED32	rw	O
6093 <sub>h</sub>	ARRAY	<i>Position factor</i>	UNSIGNED32	rw	O
6094 <sub>h</sub>	ARRAY	<i>Velocity encoder factor</i>	UNSIGNED32	rw	O
6095 <sub>h</sub>	ARRAY	<i>Velocity factor 1</i>	UNSIGNED32	rw	O
6096 <sub>h</sub>	ARRAY	<i>Velocity factor 2</i>	UNSIGNED32	rw	O
6097 <sub>h</sub>	ARRAY	<i>Acceleration factor</i>	UNSIGNED32	rw	O
607E <sub>h</sub>	VAR	<i>Polarity</i>	UNSIGNED8	rw	O

### 11.3 Object description

Objects in this group represent factors which are necessary to normalise the physical inputs and outputs. The user has to consider that the correct dimension and unit are used.

#### 11.3.1 Object 6089<sub>h</sub>: *Position notation index*

The *position notation index* is used to scale the objects for which it mandatory (see entry 'Category').

##### OBJECT DESCRIPTION

INDEX	6089 <sub>h</sub>
Name	Position notation index
Object Code	VAR
Data Type	INTEGER8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Position actual value</i></li> <li>• <i>Position demand value</i></li> <li>• <i>Target position</i></li> <li>• <i>Position window</i></li> <li>• <i>Following error window</i></li> <li>• <i>Home offset</i></li> <li>• <i>Position range limit</i></li> <li>• <i>Software position limit</i></li> <li>• <i>Target velocity</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	0

**11.3.2 Object 608A<sub>h</sub>: *Position dimension index***

## OBJECT DESCRIPTION

INDEX	608A <sub>h</sub>
Name	Position dimension index
Object Code	VAR
Data Type	UNSIGNED8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Position actual value</i></li> <li>• <i>Position demand value</i></li> <li>• <i>Target position</i></li> <li>• <i>Position window</i></li> <li>• <i>Following error window</i></li> <li>• <i>Home offset</i></li> <li>• <i>Position range limit</i></li> <li>• <i>Software position limit</i></li> <li>• <i>Target velocity</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	No

**11.3.3 Object 608B<sub>n</sub>: *Velocity notation index***

## OBJECT DESCRIPTION

INDEX	608B <sub>n</sub>
Name	Velocity notation index
Object Code	VAR
Data Type	INTEGER8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Velocity actual value</i></li> <li>• <i>Velocity demand value</i></li> <li>• <i>End velocity</i></li> <li>• <i>Profile velocity</i></li> <li>• <i>Velocity window</i></li> <li>• <i>Max profile velocity</i></li> <li>• <i>Velocity threshold</i></li> <li>• <i>Homing speeds</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	0

**11.3.4 Object 608C<sub>n</sub>: *Velocity dimension index***

## OBJECT DESCRIPTION

INDEX	608C <sub>n</sub>
Name	Velocity dimension index
Object Code	VAR
Data Type	UNSIGNED8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Velocity actual value</i></li> <li>• <i>Velocity demand value</i></li> <li>• <i>End velocity</i></li> <li>• <i>Profile velocity</i></li> <li>• <i>Velocity window</i></li> <li>• <i>Max profile velocity</i></li> <li>• <i>Velocity threshold</i></li> <li>• <i>Homing speeds</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	No

**11.3.5 Object 608D<sub>h</sub>: Acceleration notation index**

## OBJECT DESCRIPTION

INDEX	608D <sub>h</sub>
Name	Acceleration notation index
Object Code	VAR
Data Type	INTEGER8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Profile acceleration</i></li> <li>• <i>Profile deceleration</i></li> <li>• <i>Quick stop deceleration</i></li> <li>• <i>Homing acceleration</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	No

**11.3.6 Object 608E<sub>h</sub>: Acceleration dimension index**

## OBJECT DESCRIPTION

INDEX	608E <sub>h</sub>
Name	Acceleration dimension index
Object Code	VAR
Data Type	UNSIGNED8
Category	Conditional; Mandatory, if one of the following objects is supported: <ul style="list-style-type: none"> <li>• <i>Profile acceleration</i></li> <li>• <i>Profile deceleration</i></li> <li>• <i>Quick stop deceleration</i></li> <li>• <i>Homing acceleration</i></li> </ul>

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	No



**11.3.7 Object 608F<sub>h</sub>: Position encoder resolution**

The *position encoder resolution* defines the ratio of encoder increments per motor revolution.

$$\text{positionen encoder resolution} = \frac{\text{encoder increments}}{\text{motor revolutions}}$$

**OBJECT DESCRIPTION**

INDEX	608F <sub>h</sub>
Name	Position encoder resolution
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Encoder increments
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Motor revolutions
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.8 Object 6090<sub>h</sub>: Velocity encoder resolution**

The *velocity encoder resolution* defines the ratio of encoder increments/sec. per motor revolutions/sec.

$$\text{velocity encoder resolution} = \frac{\text{encoder}}{\text{motor}} \frac{\frac{\text{increments}}{\text{second}}}{\frac{\text{revolutions}}{\text{second}}}$$

**OBJECT DESCRIPTION**

INDEX	6090 <sub>h</sub>
Name	Velocity encoder resolution
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Encoder increments per second
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Motor revolutions per second
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.9 Object 6091<sub>h</sub>: Gear ratio**

The *gear ratio* defines the ratio of feed in position units per driving shaft revolutions. This includes the gear if present.

$$\text{gear ratio} = \frac{\text{motor shaft revolutions}}{\text{driving shaft revolutions}}$$

**OBJECT DESCRIPTION**

INDEX	6091 <sub>h</sub>
Name	Gear ratio
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Motor revolutions
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Shaft revolutions
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.10 Object 6092<sub>h</sub>: Feed constant**

The *feed constant* defines the ratio of feed in position units per driving shaft revolutions. This includes the gear if present.

$$\text{feed constant} = \frac{\text{feed}}{\text{driving shaft revolutions}}$$

**OBJECT DESCRIPTION**

INDEX	6092 <sub>h</sub>
Name	Feed constant
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Feed
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Shaft revolutions
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.11 Object 6093<sub>h</sub>: Position factor**

The *position factor* converts the desired position (in position units) into the internal format (in increments). This parameter may be calculated internally in the drive; nevertheless it is specified as read-writeable as the objects necessary for the calculation are defined as optional too and need not to be present in an implementation.

$$\text{position factor} = \frac{\text{position encoder resolution} \cdot \text{gear ratio}}{\text{feed constant}}$$

**OBJECT DESCRIPTION**

INDEX	6093 <sub>h</sub>
Name	Position factor
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Feed constant
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.12 Object 6094<sub>h</sub>: Velocity encoder factor**

The *velocity encoder factor* converts the desired velocity (in velocity units) into the internal format (in increments).

$$\text{velocity encoder factor} = \frac{\text{velocity encoder resolution} \cdot \text{gear ratio} \cdot \text{position unit} \cdot F_{\text{velocity (notation index)}}}{\text{feed constant} \cdot \text{velocity unit} \cdot \text{second} \cdot F_{\text{position (notation index)}}}$$

**OBJECT DESCRIPTION**

INDEX	6094 <sub>h</sub>
Name	Velocity encoder factor
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Divisor
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.13 Object 6095<sub>h</sub>: Velocity factor 1**

The *velocity factor 1* is used to convert motor data (e.g. maximum motor revolutions) into velocity data (e.g. maximum velocity), because both data items are based on different physical dimensions.

$$\text{velocity factor 1} = \frac{\text{feed constant} \cdot \text{velocity unit} \cdot \text{second} \cdot F_{\text{position unit (notation index)}}}{60 \frac{s}{\text{min}} \cdot \text{gear ratio} \cdot \text{velocity unit} \cdot F_{\text{velocity unit (notation index)}}}$$

**OBJECT DESCRIPTION**

INDEX	6095 <sub>h</sub>
Name	Velocity factor 1
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Divisor
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.14 Object 6096<sub>h</sub>: Velocity factor 2**

The *velocity factor 2* is used to convert encoder data for positions into encoder data for velocity, because both data items are based on different physical dimensions. The velocity encoder system is transformed to the position encoder.

$$\text{velocity factor 2} = \frac{\text{position encoder resolution}}{\text{velocity encoder resolution}}$$

**OBJECT DESCRIPTION**

INDEX	6096 <sub>h</sub>
Name	<i>Velocity factor 2</i>
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Divisor
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1



**11.3.15 Object 6097<sub>h</sub>: Acceleration factor**

The *acceleration factor* converts the acceleration (in acceleration unit / s) into the internal format (in increments / s).

$$\text{acceleration factor} = \frac{\text{velocity unit} \cdot \text{velocity encoder factor}}{\text{acceleration unit} \cdot \text{second}}$$

**OBJECT DESCRIPTION**

INDEX	6097 <sub>h</sub>
Name	Acceleration factor
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Divisor
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	1

**11.3.16 Object 607E<sub>h</sub>: Polarity**

*Position demand value* and *position actual value* are multiplied by 1 or -1 depending on the value of the polarity flag.

**OBJECT DESCRIPTION**

INDEX	607E <sub>h</sub>
Name	Polarity
Object Code	VAR
Data type	UNSIGNED8
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	0

**DATA DESCRIPTION**

7	6	5	4	3	2	1	0
position polarity	velocity polarity	reserved					
MSB							
							LSB

Value	Description
0	multiply by 1
1	multiply by -1

# 12 PROFILE POSITION MODE

## 12.1 General information

The overall structure for this mode is shown in Figure 15. A *target position* is applied to the trajectory generator. It is generating a *position demand value* for the position control loop described in the position control function (chapter 14). These two function blocks are optionally controlled by individual parameter sets.

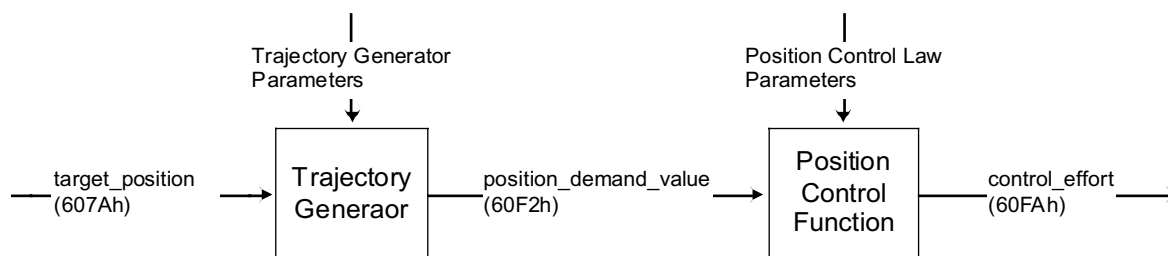
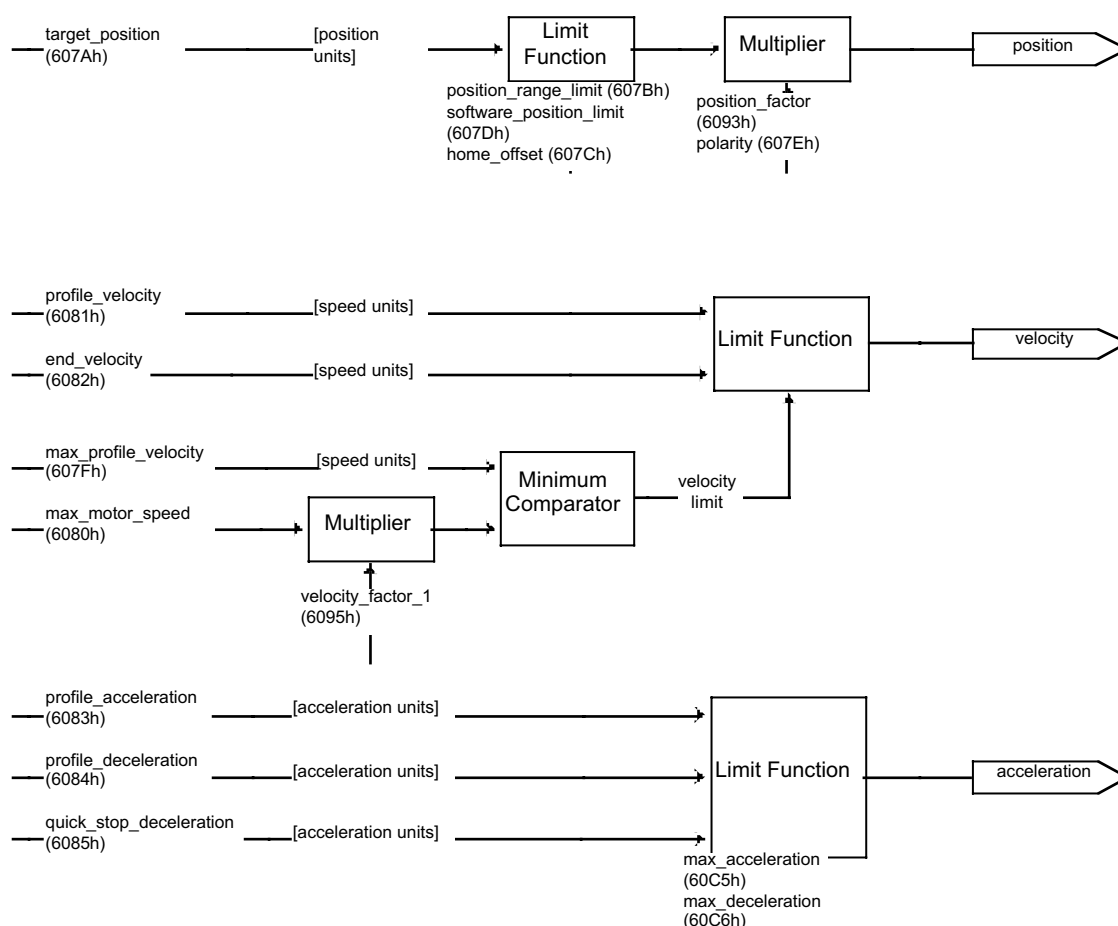


Figure 15: Overall structure for the profile position mode



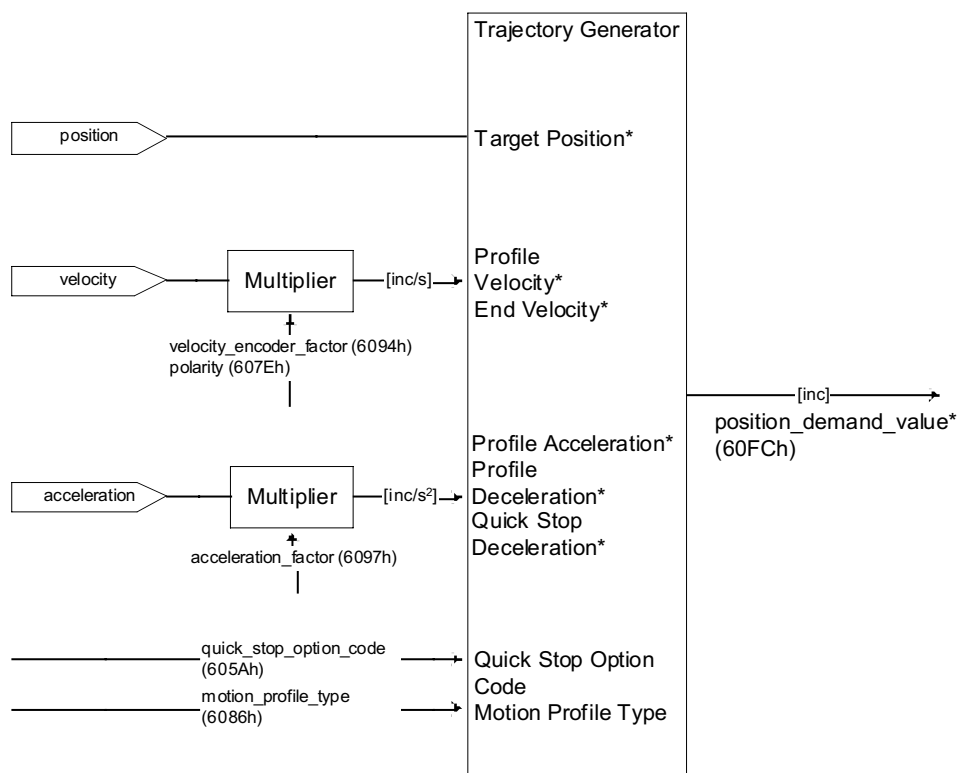


Figure 16: The trajectory generator

At the input to the trajectory generator, parameters may have optional limits applied before being normalized to internal units. Normalized parameters are denoted with an asterisk. The simplest form of a trajectory generator is just to pass through a *target position* and to transform it to a *position demand value\** with internal units (increments) only.

### 12.1.1 Input data description

Operating mode	Input parameters used
pp	<i>target position, profile velocity, end velocity, profile acceleration, profile deceleration, quick stop deceleration, position factor, quick stop option code, polarity, velocity encoder factor, motion profile type, max profile velocity, max motor speed, position range limit, software position limit, acceleration factor</i>

### 12.1.2 Output data description

The output value provided by the trajectory generator is the input for position control function. In that chapter the remotely accessible parameters of the device for a position control are described.

Operating mode	Output parameters used
pp	<i>position demand value*</i>

### 12.1.3 Internal states

The profile position mode will control by the bits of the *controlword* and *statusword*.

**12.1.3.1 Controlword of profile position mode**

15	9	8	7	6	5	4	3	0
(see 10.3.1)	Halt	(see 10.3.1)	abs / rel	Change set immediately	New set-point	(see 10.3.1)		
MSB							LSB	

Name	Value	Description
New set-point	0	Does not assume <i>target position</i>
	1	Assume <i>target position</i>
Change set immediately	0	Finish the actual positioning and then start the next positioning
	1	Interrupt the actual positioning and start the next positioning
abs / rel	0	<i>Target position</i> is an absolute value
	1	<i>Target position</i> is a relative value
Halt	0	Execute positioning
	1	Stop axle with <i>profile deceleration</i> (if not supported with <i>profile acceleration</i> )

Table 9: Profile position mode bits of the *controlword***12.1.3.2 Statusword of profile position mode**

15	14	13	12	11	10	9	0
(see 10.3.2)	Following error	Set-point acknowledge	(see 10.3.2)	Target reached	(see 10.3.2)		
MSB							LSB

Name	Value	Description
Target reached	0	Halt = 0: <i>Target position</i> not reached Halt = 1: Axle decelerates
	1	Halt = 0: <i>Target position</i> reached Halt = 1: Velocity of axle is 0
Set-point acknowledge	0	Trajectory generator has not assumed the positioning values (yet)
	1	Trajectory generator has assumed the positioning values
Following error	0	No following error
	1	Following error

Table 10: Profile position mode bits of the *statusword***12.2 Object dictionary entries****12.2.1 Objects defined in this chapter**

Index	Object	Name	Type	Attr.	M/O
607A <sub>h</sub>	VAR	<i>Target position</i>	INTEGER32	rw	M
607B <sub>h</sub>	ARRAY	<i>Position range limit</i>	INTEGER32	rw	O
607D <sub>h</sub>	ARRAY	<i>Software position limit</i>	INTEGER32	rw	O
607F <sub>h</sub>	VAR	<i>Max profile velocity</i>	UNSIGNED32	rw	O

Index	Object	Name	Type	Attr.	M/O
6080 <sub>h</sub>	VAR	<i>Max motor speed</i>	UNSIGNED32	rw	O
6081 <sub>h</sub>	VAR	<i>Profile velocity</i>	UNSIGNED32	rw	M
6082 <sub>h</sub>	VAR	<i>End velocity</i>	UNSIGNED32	rw	O
6083 <sub>h</sub>	VAR	<i>Profile acceleration</i>	UNSIGNED32	rw	M
6084 <sub>h</sub>	VAR	<i>Profile deceleration</i>	UNSIGNED32	rw	O
6085 <sub>h</sub>	VAR	<i>Quick stop deceleration</i>	UNSIGNED32	rw	O
6086 <sub>h</sub>	VAR	<i>Motion profile type</i>	INTEGER16	rw	M
60C5 <sub>h</sub>	VAR	<i>Max acceleration</i>	UNSIGNED32	rw	O
60C6 <sub>h</sub>	VAR	<i>Max deceleration</i>	UNSIGNED32	rw	O

### 12.2.2 Objects defined in other chapters

Index	Object	Name	Type	Chapter
6040 <sub>h</sub>	VAR	<i>Controlword</i>	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	<i>Statusword</i>	UNSIGNED16	dc
605A <sub>h</sub>	VAR	<i>Quick stop option code</i>	INTEGER16	dc
607E	VAR	<i>Polarity</i>	UNSIGNED8	fg
6093 <sub>h</sub>	ARRAY	<i>Position factor</i>	UNSIGNED32	fg
6094 <sub>h</sub>	ARRAY	<i>Velocity encoder factor</i>	UNSIGNED32	fg
6095 <sub>h</sub>	ARRAY	<i>Velocity factor 1</i>	UNSIGNED32	fg
6097 <sub>h</sub>	ARRAY	<i>Acceleration factor</i>	UNSIGNED32	fg

## 12.3 Object description

### 12.3.1 Object 607A<sub>h</sub>: Target position

The *target position* is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, *motion profile type* etc. The *target position* is given in user defined position units. It is converted to position increments using the *position factor* (see chapter 11). The *target position* will be interpreted as absolute or relative depending on the 'abs / rel' flag in the *controlword*.

#### OBJECT DESCRIPTION

INDEX	607A <sub>h</sub>
Name	Target position
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pp or pc supported

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**12.3.2 Object 607B<sub>h</sub>: Position range limit**

*Position range limit* contains two sub-parameters, *min position range limit* and *max position range limit*. These limit the numerical range of the input value. On reaching or exceeding these limits, the input value automatically wraps to the other end of the range. Wrap-around of the input value can be prevented by setting software position limits.

**OBJECT DESCRIPTION**

INDEX	607B <sub>h</sub>
Name	Position range limit
Object Code	ARRAY
Data Type	INTEGER32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Min position range limit
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	-2 <sup>31</sup>

Sub-Index	2
Description	Max position range limit
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	2 <sup>31</sup> -1

**12.3.3 Object 607D<sub>n</sub>: Software position limit**

*Software position limit* contains the sub-parameters *min position limit* and *max position limit*. These parameters define the absolute position limits for the *position demand value* and the *position actual value*. Every new *target position* must be checked against these limits. The limit positions are specified in position units (same as *target position*) and are always relative to the machine home position. Before being compared with the *target position* they must be corrected internally by the *home offset* as follows:

$$\begin{aligned}\text{corrected min position limit} &= \text{min position limit} - \text{home offset} \\ \text{corrected max position limit} &= \text{max position limit} - \text{home offset}\end{aligned}$$

This calculation needs only be performed when *home offset* or *software position limit* is changed.

**OBJECT DESCRIPTION**

INDEX	607D <sub>n</sub>
Name	Software position limit
Object Code	ARRAY
Data Type	INTEGER32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Min position limit
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	-2 <sup>31</sup>

Sub-Index	2
Description	Max position limit
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	2 <sup>31</sup> -1



**12.3.4 Object 607F<sub>h</sub>: Max profile velocity**

The *max profile velocity* is the maximum allowed speed in either direction during a profiled move. It is given in the same units as *profile velocity*.

**OBJECT DESCRIPTION**

INDEX	607F <sub>h</sub>
Name	Max profile velocity
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.5 Object 6080<sub>h</sub>: Max motor speed**

The *max motor speed* is the maximum allowable speed for the motor in either direction and is given in rpm. This is used to protect the motor and can be taken from the motor data sheet.

**OBJECT DESCRIPTION**

INDEX	6080 <sub>h</sub>
Name	Max motor speed
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.6 Object 6081<sub>h</sub>: Profile velocity**

The *profile velocity* is the velocity normally attained at the end of the acceleration ramp during a profiled move and is valid for both directions of motion. The *profile velocity* is given in user defined speed units. It is converted to position increments per second using the *velocity encoder factor* (see chapter 11).

**OBJECT DESCRIPTION**

INDEX	6081 <sub>h</sub>
Name	Profile velocity
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional; Mandatory, if pp or pv supported

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.7 Object 6082<sub>h</sub>: End velocity**

The *end velocity* defines the velocity which the drive must have on reaching the *target position*. Normally, the drive stops at the *target position*, i.e. the *end velocity* = 0. The *end velocity* is given in the same units as *profile velocity*.

## OBJECT DESCRIPTION

INDEX	6082 <sub>h</sub>
Name	End velocity
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## VALUE DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

**12.3.8 Object 6083<sub>h</sub>: Profile acceleration**

The *profile acceleration* is given in user defined acceleration units. It is converted to position increments per second<sup>2</sup> using the normalizing factors (see chapter 11).

## OBJECT DESCRIPTION

INDEX	6083 <sub>h</sub>
Name	Profile acceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional; Mandatory, if pp or pv supported

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.9 Object 6084<sub>h</sub>: Profile deceleration**

The *profile deceleration* is given in the same units as *profile acceleration*. If this parameter is not supported, then the *profile acceleration* value is also used for deceleration.

**OBJECT DESCRIPTION**

INDEX	6084 <sub>h</sub>
Name	Profile deceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.10 Object 6085<sub>h</sub>: Quick stop deceleration**

The *quick stop deceleration* is the deceleration used to stop the motor if the 'Quick Stop' command is given and the *quick stop option code* (see 605A<sub>h</sub>) is set to 2. The *quick stop deceleration* is given in the same units as the *profile acceleration*.

**OBJECT DESCRIPTION**

INDEX	6085 <sub>h</sub>
Name	Quick stop deceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.11 Object 6086<sub>h</sub>: Motion profile type**

The *motion profile type* is used to select the type of motion profile used to perform a profiled move.

Value	Description
-32768 ... -1	manufacturer specific
0	Linear ramp (trapezoidal profile)
1	sin <sup>2</sup> ramp
2	Jerk-free ramp
3	Jerk-limited ramp
4 .. 32767	reserved

## OBJECT DESCRIPTION

INDEX	6086 <sub>h</sub>
Name	Motion profile type
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if pp or pv supported

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**12.3.12 Object 60C5<sub>h</sub>: Max acceleration**

To prevent the motor and the application from being destroyed, the *max acceleration* can be used to limit the acceleration to an acceptable value.

The *max acceleration* is given in user defined *acceleration units* (608D<sub>h</sub>, 608E<sub>h</sub>). It is converted to position increments per second<sup>2</sup> using the *acceleration factor* (6097<sub>h</sub>).

## OBJECT DESCRIPTION

INDEX	60C5 <sub>h</sub>
Name	Max acceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**12.3.13 Object 60C6<sub>h</sub>: Max deceleration**

To prevent the motor and the application from being destroyed, the *max deceleration* can be used to limit the deceleration to an acceptable value.

The *max deceleration* is given in the same units as the *max acceleration* (60C5<sub>h</sub>). If this parameter is not supported, then the *max acceleration* value is also used for deceleration.

## OBJECT DESCRIPTION

INDEX	60C6 <sub>h</sub>
Name	Max deceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

## 12.4 Functional description

Two different ways to apply *target positions* to a drive, are supported by this device profile.

## Set of set-points:

After reaching the *target position* the drive unit immediately processes the next *target position* which results in a move where the velocity of the drive normally is not reduced to zero after achieving a set-point.

## Single set-point:

After reaching the *target position* the drive unit signals this status to a host computer and then receives a new set-point. After reaching a *target position* the velocity normally is reduced to zero before starting a move to the next set-point.

The two modes are controlled by the timing of the bits '*new set-point*' and '*change set immediately*' in the *controlword* and '*set-point acknowledge*' in the *statusword*. These bits allow to set up a request-response mechanism in order to prepare a set of set-points while another set still is processed in the drive unit. This minimizes reaction times within a control program on a host computer.

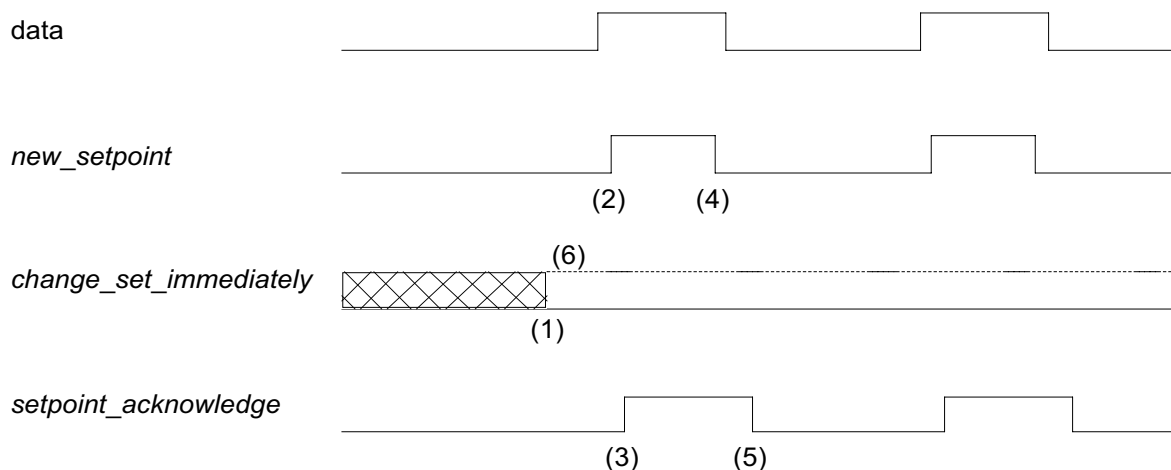


Figure 17: Set-point transmission from a host computer

Figure 17, Figure 18 and Figure 19 show the difference between the "set of set-points" mode and the "single set-point" mode. The initial status of the bit '*change set immediately*' in the *controlword* determines which mode is used. To keep simple these examples, only trapezoidal moves are used.

If the bit '*change set immediately*' is "0" (continuously drawn line in Figure 17) a single set-point is expected by the drive (1). After data is applied to the drive, a host signals that the data is valid by changing the bit '*new set-point*' to "1" in the *controlword* (2). The drive responds with '*set-point acknowledge*' set to "1" in the *statusword* (3) after it recognized and buffered the new valid data. Now the host may release '*new set-point*' (4) and afterwards the drive signals with '*set-point acknowledge*' equal "0" its ability to accept new data again (5). In Figure 18 this mechanism results in a velocity of zero after ramping down in order to reach a *target position*  $x_1$  at  $t_1$ . After signaling to the host, that the set-point is reached like described above, the next *target position*  $x_2$  is processed at  $t_2$  and reached at  $t_3$ .

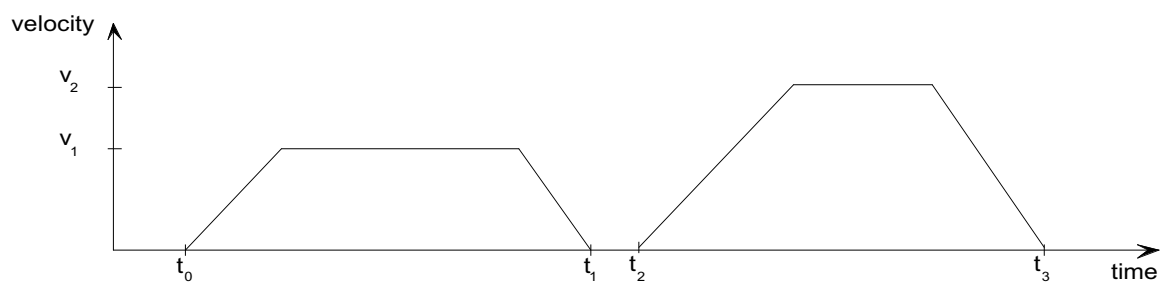


Figure 18: Single set-point

If the bit 'change set immediately' is "1" (dashed line in Figure 17) the new target position will be active immediately. In Figure 19 the drive receives the first *target position* at  $t_0$ . At the time point  $t_1$  the drive receives a second *target position*. The drive readapts the actual move to the new target position immediately.

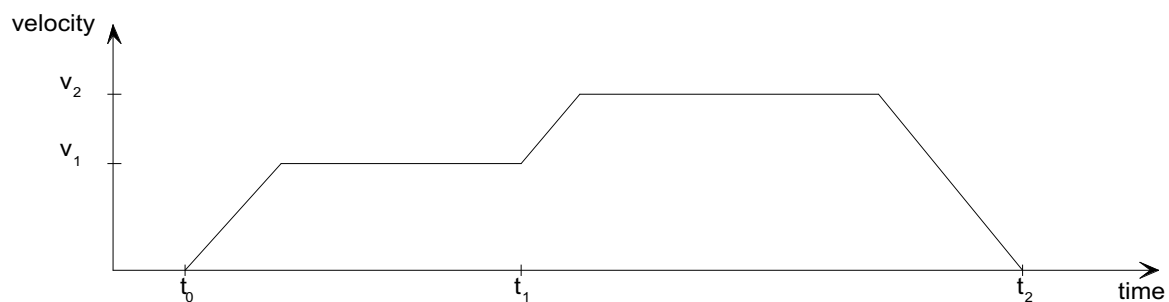


Figure 19: Change set immediately

# 13 HOMING MODE

## 13.1 General information

This chapter describes the method by which a drive seeks the home position (also called, the datum, reference point or zero point). There are various methods of achieving this using limit switches at the ends of travel or a home switch (zero point switch) in mid-travel, most of the methods also use the index (zero) pulse train from an incremental encoder.

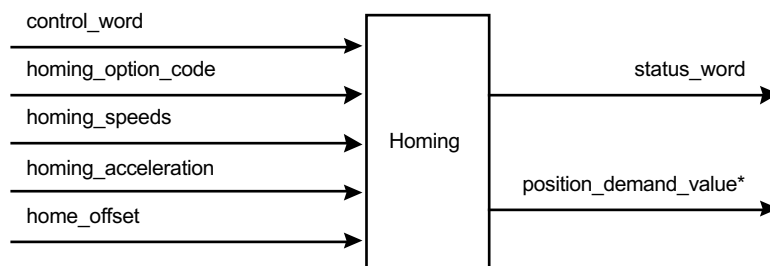


Figure 20: The homing function

### 13.1.1 Input data description

The user can specify the speeds, acceleration and the method of homing. There is a further object *home offset* which allows the user to displace zero in the user's coordinate system from the home position.

There are two *homing speeds*; in a typical cycle the faster speed is used to find the home switch and the slower speed is used to find the index pulse. The manufacturer is allowed some discretion in the use of these speeds as the response to the signals may be dependent upon the hardware used.

### 13.1.2 Output data description

There is no output data except for those bits in the *statusword* which return the status or result of the homing process and the demand to the position control loops.

### 13.1.3 Internal states

The homing mode will control by the bits of the *controlword* and *statusword*.

#### 13.1.3.1 Controlword of homing mode

15	9	8	7	6	5	4	3	0
(see 10.3.1)		Halt	(see 10.3.1)		reserved	Homing operation start		(see 10.3.1)
MSB						LSB		

Name	Value	Description
Homing operation start	0	Homing mode inactive
	0 → 1	Start homing mode
	1	Homing mode active
	1 → 0	Interrupt homing mode
Halt	0	Execute the instruction of bit 4
	1	Stop axle with homing acceleration

Table 11: Homing mode bits of the *controlword*

**13.1.3.2 Statusword of homing mode**

15	14	13	12	11	10	9	0
(see 10.3.2)	Homing error	Homing attained	(see 10.3.2)	Target reached	(see 10.3.2)		
MSB				LSB			

Name	Value	Description
Target reached	0	Halt = 0: Home position not reached Halt = 1: Axle decelerates
	1	Halt = 0: Home position reached Halt = 1: Axle has velocity 0
Homing attained	0	Homing mode not yet completed
	1	Homing mode carried out successfully
Homing error	0	No homing error
	1	Homing error occurred; Homing mode carried out not successfully; The error cause is found by reading the error code

Table 12: Homing mode bits of the *statusword***13.2 Object dictionary entries****13.2.1 Objects defined in this chapter**

Index	Object	Name	Type	Attr.	M/O
607C <sub>h</sub>	VAR	<i>Home offset</i>	INTEGER32	rw	O
6098 <sub>h</sub>	VAR	<i>Homing method</i>	INTEGER8	rw	M
6099 <sub>h</sub>	ARRAY	<i>Homing speeds</i>	UNSIGNED32	rw	M
609A <sub>h</sub>	VAR	<i>Homing acceleration</i>	UNSIGNED32	rw	O

**13.2.2 Objects defined in other chapters**

Index	Object	Name	Type	Chapter
6040 <sub>h</sub>	VAR	<i>Controlword</i>	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	<i>Statusword</i>	UNSIGNED16	dc

**13.3 Object description****13.3.1 Object 607C<sub>h</sub>: Home offset**

The *home offset* object is the difference between the zero position for the application and the machine home position (found during homing), it is measured in position units. During homing the machine home position is found and once the homing is completed the zero position is offset from the home position by adding the *home offset* to the home position. All subsequent absolute moves shall be taken relative to this new zero position. This is illustrated in the following diagram.



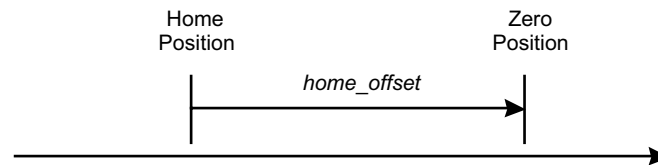


Figure 21: Home offset

If the *home offset* is not implemented then it shall be zero.

#### OBJECT DESCRIPTION

INDEX	607C <sub>h</sub>
Name	Home offset
Object Code	VAR
Data Type	INTEGER32
Category	Optional

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	0

### 13.3.2 Object 6098<sub>h</sub>: Homing method

The *homing method* object determines the method that will be used during homing.

#### OBJECT DESCRIPTION

INDEX	6098 <sub>h</sub>
Name	Homing method
Object Code	VAR
Data Type	INTEGER8
Category	Conditional; Mandatory, if hm supported

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER8
Default Value	0

#### DATA DESCRIPTION

Value	Description
-128 .. -1	manufacturer specific
0	No homing operation required
1..35	Methods 1 to 35 (see the functional description)
36 .. 127	reserved

**13.3.3 Object 6099<sub>h</sub>: Homing speeds**

This entry in the object dictionary defines the speeds used during homing and is given velocity units.

**OBJECT DESCRIPTION.**

INDEX	6099 <sub>h</sub>
Name	Homing speeds
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Mandatory, if hm supported

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Speed during search for switch
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

Sub-Index	2
Description	Speed during search for zero
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

### 13.3.4 Object 609A<sub>n</sub>: *Homing acceleration*

The *homing acceleration* establishes the acceleration to be used for all accelerations and decelerations with the standard homing modes and is given in acceleration units.

#### OBJECT DESCRIPTION

INDEX	609A <sub>n</sub>
Name	Homing acceleration
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

## 13.4 Functional description

By choosing a method of homing by writing a value to *homing method* will clearly establish

- the homing signal (positive limit switch, negative limit switch, home switch)
- the direction of actuation and where appropriate
- the position of the index pulse.

The home position and the zero position are offset by the *home offset*, see the definition of *home offset* for how this offset is used.

Various homing positions are illustrated in the following diagrams. An encircled number indicates the code for selection of this homing position. The direction of movement is also indicated. Further homing methods may be defined by the manufacturer using the negative values of *homing method*.

There are four sources of homing signal available, these are the negative and positive limit switches, the home switch and the index pulse from an encoder.

In the diagrams of homing sequences shown below, the encoder count increases as the axle's position moves to the right, in other words the left is the minimum position and the right is the maximum position.

For the operation of positioning drives, an exact knowledge of the absolute position is normally required. Since for cost reasons, drives often do not have an absolute encoder, a homing operation is necessary. There are several, application-specific methods. The *homing method* is used for selection.

The exact sequence of the homing operation is clearly described by the method. In some circumstances, a device has several methods to choose from, using the *homing method*.

### 13.4.1 Homing methods

The following sub-sections describe the details of how each of the homing modes shall function.

#### 13.4.1.1 Method 1: Homing on the negative limit switch and index pulse

Using this method the initial direction of movement is leftward if the negative limit switch is inactive (here shown as low). The home position is at the first index pulse to the right of the position where the negative limit switch becomes inactive.

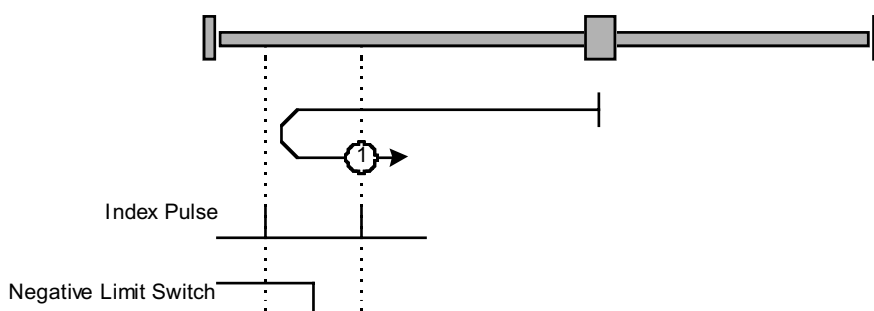


Figure 22: Homing on the negative limit switch and index pulse

#### 13.4.1.2 Method 2: Homing on the positive limit switch and index pulse

Using this method the initial direction of movement is rightward if the positive limit switch is inactive (here shown as low). The position of home is at the first index pulse to the left of the position where the positive limit switch becomes inactive.

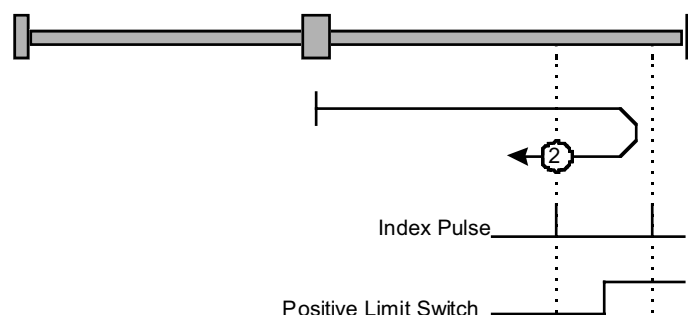


Figure 23: Homing on the positive limit switch and index pulse

#### 13.4.1.3 Methods 3 and 4: Homing on the positive home switch and index pulse

Using methods 3 or 4 the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

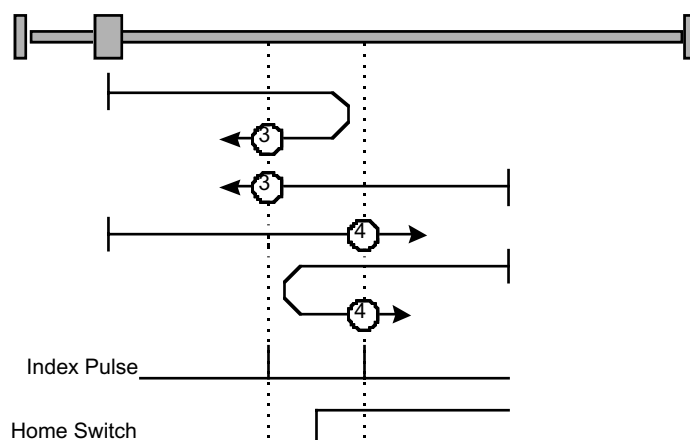


Figure 24: Homing on the positive home switch and index pulse

#### 13.4.1.4 Methods 5 and 6: Homing on the negative home switch and index pulse

Using methods 5 or 6 the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either the left or the right of the point where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

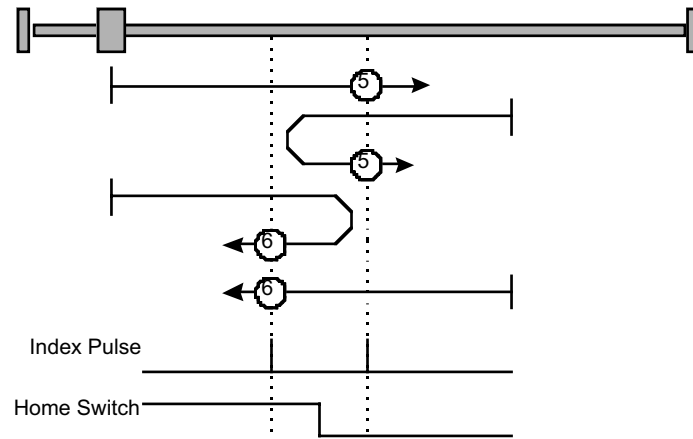


Figure 25: Homing on the negative home switch and index pulse

#### 13.4.1.5 Methods 7 to 14: Homing on the home switch and index pulse

These methods use a home switch which is active over only portion of the travel, in effect the switch has a 'momentary' action as the axle's position sweeps past the switch.

Using methods 7 to 10 the initial direction of movement is to the right, and using methods 11 to 14 the initial direction of movement is to the left except if the home switch is active at the start of the motion. In this case the initial direction of motion is Dependent on the edge being sought. The home position is at the index pulse on either side of the rising or falling edges of the home switch, as shown in the following two diagrams. If the initial direction of movement leads away from the home switch, the drive must reverse on encountering the relevant limit switch.

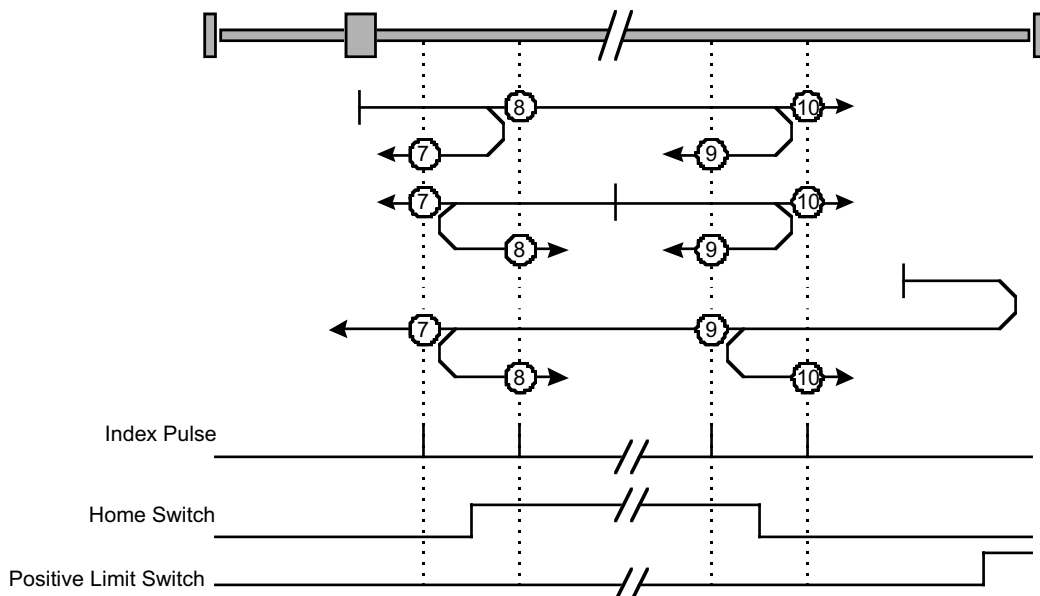


Figure 26: Homing on the home switch and index pulse - positive initial move

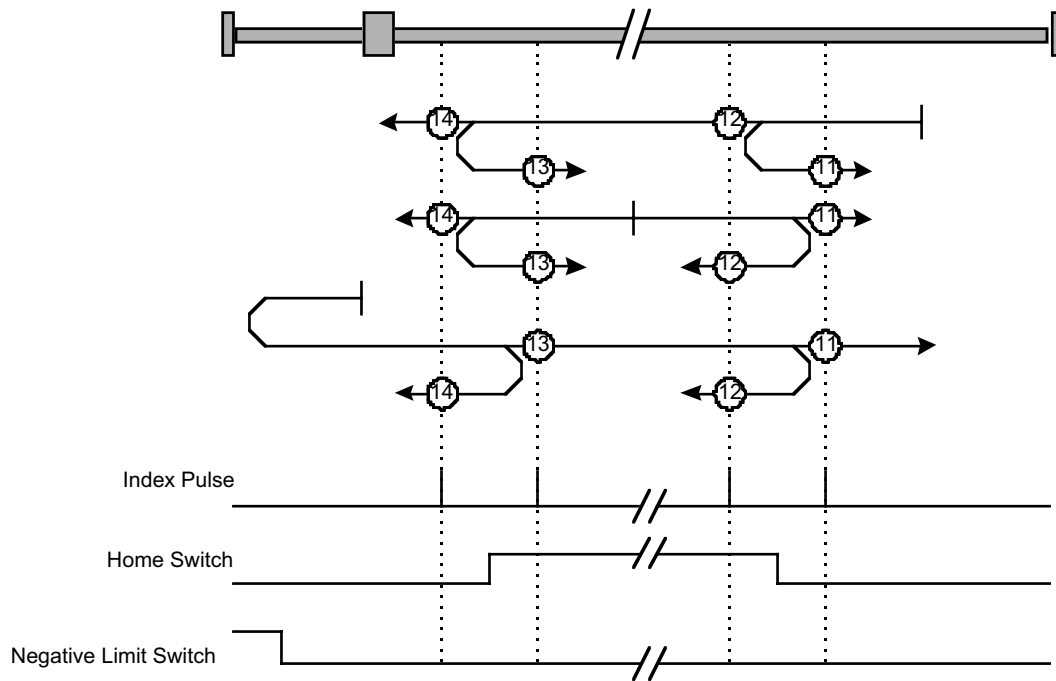


Figure 27: Homing on the home switch and index pulse - negative initial move

#### 13.4.1.6 Methods 15 and 16: Reserved

These methods are reserved for future expansion of the homing mode.

#### 13.4.1.7 Methods 17 to 30: Homing without an index pulse

These methods are similar to methods 1 to 14 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. For example methods 19 and 20 are similar to methods 3 and 4 as shown in the following diagram.

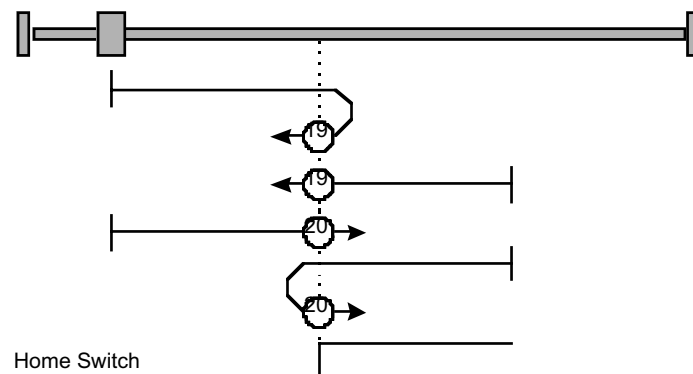


Figure 28: Homing on the positive home switch

#### 13.4.1.8 Methods 31 and 32: Reserved

These methods are reserved for future expansion of the homing mode.

**13.4.1.9 Methods 33 to 34: Homing on the index pulse**

Using methods 33 or 34 the direction of homing is negative or positive respectively. The home position is at the index pulse found in the selected direction.

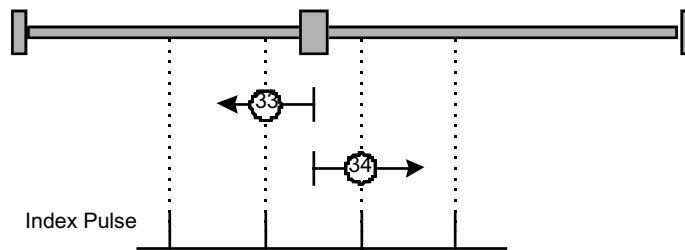


Figure 29: Homing on the index pulse

**13.4.1.10 Method 35: Homing on the current position**

In method 35 the current position is taken to be the home position.

# 14 POSITION CONTROL FUNCTION

## 14.1 General information

In this chapter, all parameters are described which are necessary for a closed loop position control. The control loop is fed with the *position demand value* as one of the outputs of the trajectory generator and with the output of the position detection unit (*position actual value*) like a resolver or encoder as input parameters. The behavior of the control may be influenced by control parameters which are externally applicable. To keep stable the loop, a relative limitation of the output using the previous *control effort* is possible. In order not to exceed physical limits of a drive, an absolute limit function is implemented for the *control effort*. The *control effort* may be a *velocity demand value*, a *position demand value* or any other output value, depending on the modes of operation implemented by a manufacturer. Especially in cascaded control structures, where a position control is followed by a torque control, e.g. the *control effort* of the position control loop is used as an input for a further calculation.

All values are transformed - if necessary - from user defined units to normalised units like increments with the functions of the chapter 11.

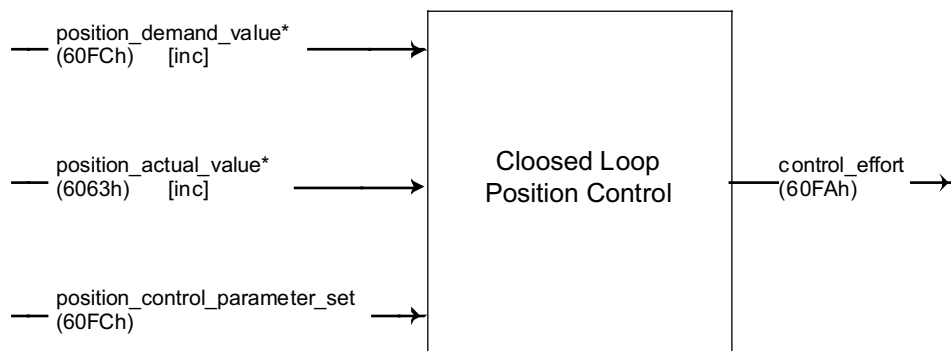


Figure 30: Position control function

Within this chapter, the following sub-functions are defined:

### 14.1.1 Following error

A *position actual value* outside the allowed range of the *following error window* around a *position demand value* for longer than the *following error time out* results in setting bit 13 *following error* in the *statusword*.

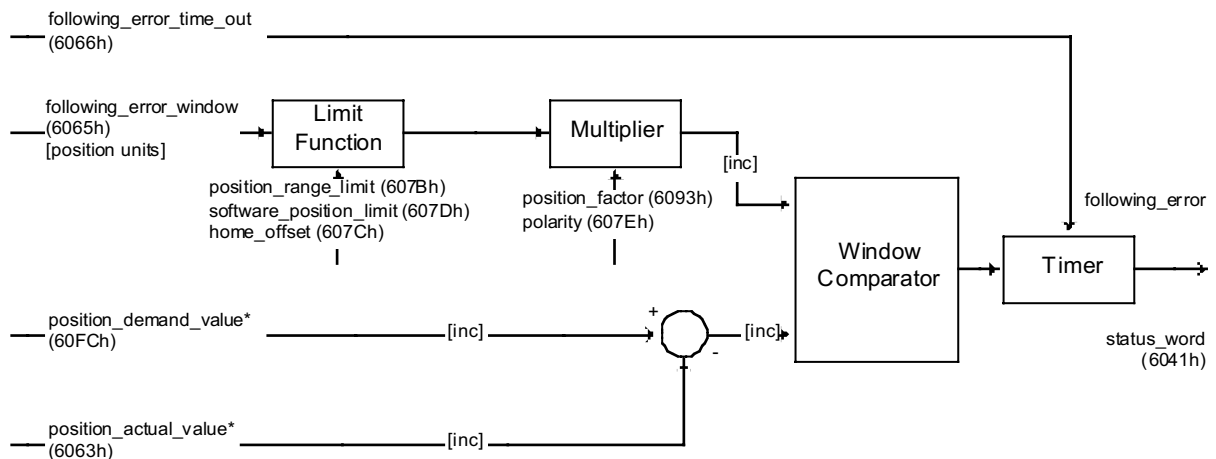


Figure 31: Following error - functional overview



### 14.1.2 Position reached

This function offers the possibility to define a position range around a *position demand value* to be regarded as valid. If a drives position is within this area for a specified time - the *position window time* - the related control bit 10 *target reached* in the *statusword* is set.

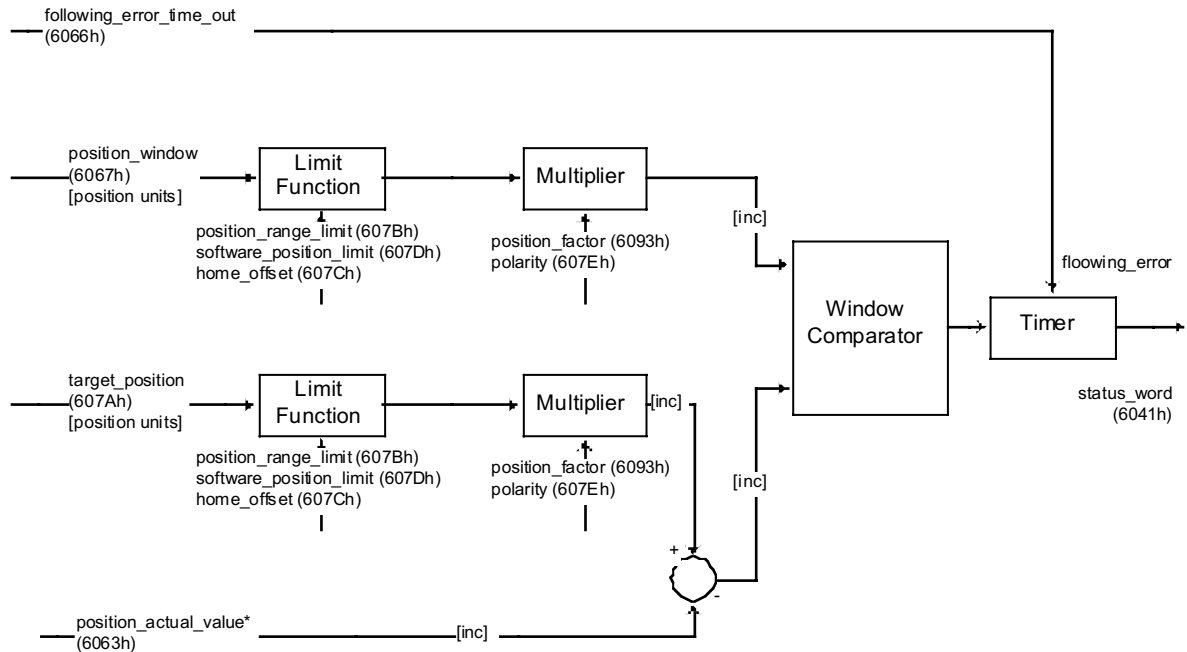


Figure 32: Position reached - functional overview

The control functions following error and position reached have direct access to the *statusword* and give immediate notification to the user if their results change.

### 14.1.3 Input data description

Depending on the supported modes of operation and on the capabilities of different categories of drives, only some of the mentioned input parameters may be necessary.

Operating mode	Input parameters used
pp, hm, ip	<i>position demand value*</i> , <i>position window time</i> , <i>position window</i> , <i>following error time out</i> , <i>following error window</i> , <i>position actual value</i> , <i>digital inputs</i> , <i>target position</i> , <i>position factor</i> , <i>position range limit</i> , <i>polarity</i>

### 14.1.4 Output data description

Operating mode	Output parameters used
pp, hm, ip	<i>statusword</i> , <i>control effort</i> , <i>digital outputs</i>

## 14.2 Object dictionary entries

### 14.2.1 Objects defined in this chapter

Index	Object	Name	Type	Attr.	M/O
6062 <sub>h</sub>	VAR	<i>Position demand value</i>	INTEGER32	ro	O
6063 <sub>h</sub>	VAR	<i>Position actual value*</i>	INTEGER32	ro	O
6064 <sub>h</sub>	VAR	<i>Position actual value</i>	INTEGER32	ro	M
6065 <sub>h</sub>	VAR	<i>Following error window</i>	UNSIGNED32	rw	O

Index	Object	Name	Type	Attr.	M/O
6066 <sub>h</sub>	VAR	<i>Following error time out</i>	UNSIGNED16	rw	O
6067 <sub>h</sub>	VAR	<i>Position window</i>	UNSIGNED32	rw	O
6068 <sub>h</sub>	VAR	<i>Position window time</i>	UNSIGNED16	rw	O
60F4 <sub>h</sub>	VAR	<i>Following error actual value</i>	INTEGER32	ro	O
60FA <sub>h</sub>	VAR	<i>Control effort</i>	INTEGER32	ro	O
60FB <sub>h</sub>	RECORD	<i>Position control parameter set</i>	(manufacturer specific)	rw	O
60FC <sub>h</sub>	VAR	<i>Position demand value*</i>	INTEGER32	ro	O

#### 14.2.2 Objects defined in other chapters

Index	Object	Name	Type	Chapter
607A <sub>h</sub>	VAR	<i>Target position</i>	INTEGER32	pp
607B <sub>h</sub>	VAR	<i>Position range limit</i>	INTEGER32	pp
607C <sub>h</sub>	VAR	<i>Home offset</i>	INTEGER32	hm
607D <sub>h</sub>	VAR	<i>Software position limit</i>	INTEGER32	pp
607E <sub>h</sub>	VAR	<i>Polarity</i>	UNSIGNED8	fg
6093 <sub>h</sub>	VAR	<i>Position factor</i>	UNSIGNED32	fg
6094 <sub>h</sub>	ARRAY	<i>Velocity encoder factor</i>	UNSIGNED32	fg
6095 <sub>h</sub>	ARRAY	<i>Velocity factor 1</i>	UNSIGNED32	fg
6097 <sub>h</sub>	ARRAY	<i>Acceleration factor</i>	UNSIGNED32	fg
6041 <sub>h</sub>	VAR	<i>Controlword</i>	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	<i>Statusword</i>	UNSIGNED16	dc

### 14.3 Object description

#### 14.3.1 Object 6062<sub>h</sub>: Position demand value

The *position demand value* is given in position units.

##### OBJECT DESCRIPTION

INDEX	6062 <sub>h</sub>
Name	Position demand value
Object Code	VAR
Data Type	INTEGER32
Category	Optional

##### ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**14.3.2 Object 6063<sub>h</sub>: Position actual value\***

The actual value of the position measurement device is one of the two input values of the closed loop position control. The data unit is defined as increments. If necessary, the data unit must be transformed with the *position factor* defined in chapter 11 from user defined units to increments.

## OBJECT DESCRIPTION

INDEX	6063 <sub>h</sub>
Name	Position actual value*
Object Code	VAR
Data Type	INTEGER32
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**14.3.3 Object 6064<sub>h</sub>: Position actual value**

This object represents the actual value of the position measurement device in user defined units.

## OBJECT DESCRIPTION

INDEX	6064 <sub>h</sub>
Name	Position actual value
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pc supported Optional, if pp, ip, hm or tq supported

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**14.3.4 Object 6065<sub>h</sub>: Following error window**

The *following error window* defines a range of tolerated position values symmetrically to the *position demand value*. As it is in most cases used with user defined units, a transformation into increments with the *position factor* is necessary. If the *position actual value* is out of the *following error window*, a following error occurs

A following error might occur when

- a drive is blocked,
- unreachable profile velocity occurs, or
- at wrong closed loop coefficients.

If the value of the following error window is  $2^{32}-1$ , the following control is switched off.

## OBJECT DESCRIPTION

INDEX	6065 <sub>h</sub>
Name	Following error window
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**14.3.5 Object 6066<sub>h</sub>: Following error time out**

When a following error occurs longer than the defined value of the time-out given in multiples of milliseconds, the corresponding bit 13 *following error* in the *statusword* will be set to one. The reaction of the drive when a following error occurs, is manufacturer specific.

## OBJECT DESCRIPTION

INDEX	6066 <sub>h</sub>
Name	Following error time out
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**14.3.6 Object 6067<sub>h</sub>: Position window**

The *position window* defines a symmetrical range of accepted positions relatively to the *target position*. If the actual value of the position encoder is within the *position window*, this *target position* is regarded as reached. As the user mostly prefers to specify the *position window* in his application in user defined units, the *position factor* of chapter 11 must be used to transform this value into increments. The *target position* has to be handled in the same manner as in the Trajectory Generator concerning limiting functions and transformation into internal machine units before it can be used with this function.

If the value of the position window is  $2^{32}-1$ , the position window control is switched off.

## OBJECT DESCRIPTION

INDEX	6067 <sub>h</sub>
Name	Position window
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**14.3.7 Object 6068<sub>h</sub> : Position window time**

When the actual position is within the *position window* during the defined *position window time* which is given in multiples of milliseconds, the corresponding bit 10 *target reached* in the *statusword* will be set to one.

## OBJECT DESCRIPTION

INDEX	6068 <sub>h</sub>
Name	Position window time
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**14.3.8 Object 60F4<sub>h</sub> : Following error actual value**

This object represents the actual value of the following error, it is given in user defined position units.

## OBJECT DESCRIPTION

INDEX	60F4 <sub>h</sub>
Name	Following error actual value
Object Code	VAR
Data Type	INTEGER32
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**14.3.9 Object 60FA<sub>n</sub>: Control effort**

The output of the position control loop is the *control effort*. It is particular to the Position control function that the notation of the *control effort* is mode dependent and therefore not specified in the object description.

**OBJECT DESCRIPTION**

INDEX	60FA <sub>n</sub>
Name	Control effort
Object Code	VAR
Data Type	INTEGER32
Category	Optional

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**14.3.10 Object 60FB<sub>n</sub>: Position control parameter set**

In order to control the behavior of the position control loop one or more parameters are necessary. This object is a means to define control parameters which are highly manufacturer specific. For this reason, these parameters shall not be described in this document at all.

**OBJECT DESCRIPTION**

INDEX	60FB <sub>n</sub>
Name	Position control parameter set
Object Code	RECORD
Data Type	manufacturer specific
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 .. 254
Default Value	No

Sub-Index	1
Description	manufacturer specific
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	manufacturer specific
Default Value	No

Sub-Index	2
Description	manufacturer specific
Entry Category	Optional
Access	rw
PDO Mapping	Possible
Value Range	manufacturer specific
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry Category	Optional
Access	rw
PDO Mapping	Possible
Value Range	manufacturer specific
Default Value	No

#### 14.3.11 Object 60FC<sub>n</sub>: *Position demand value*\*

This output of the trajectory generator in profile position mode is an internal value using increments as unit what is expressed with an \*. To save calculation time for some applications, this object is additionally introduced to the *position demand value* (6062<sub>n</sub>).

##### OBJECT DESCRIPTION

INDEX	60FC <sub>n</sub>
Name	Position demand value*
Object Code	VAR
Data Type	INTEGER32
Category	Optional

##### ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

#### 14.4 Functional description

Figure 33 shows the meaning of the sub-function *position reached*. Symmetrically around the *target position* a window is defined for the accepted position range. If a drive is situated in the accepted position range over the time *position window time* the bit *target reached* (bit 10) in the *statusword* is to set.

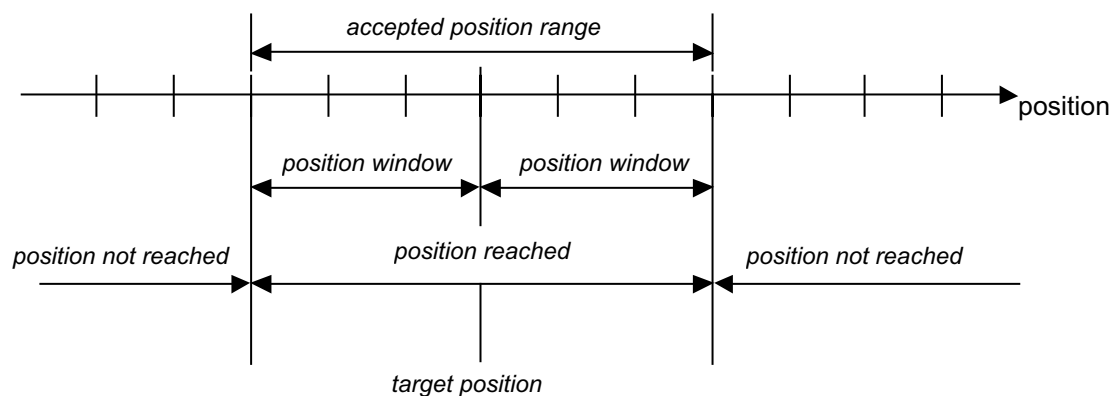


Figure 33: Position reached

Figure 34 shows the meaning of the sub-function *following error* in the profile position mode. Symmetrically around the reference position a window is defined for the accepted following error tolerance. If a drive is situated out of the accepted position range for more than *following error time out* the bit *following error* (bit 13) in the *statusword* is set.

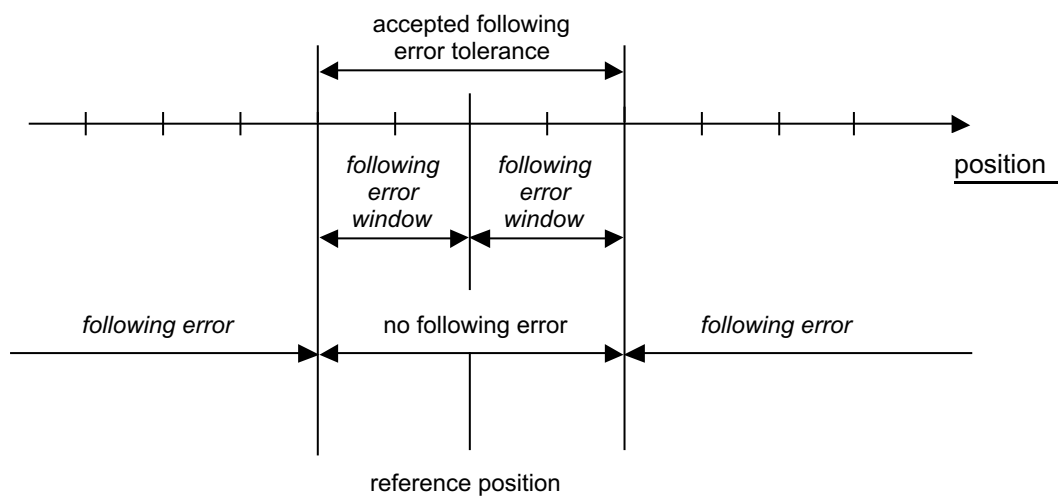


Figure 34: Following error



# 15 INTERPOLATED POSITION MODE

## 15.1 General information

The interpolated position mode is used to control multiple coordinated axes or a single axle with the need for time-interpolation of set-point data. The interpolated position mode normally uses time synchronization mechanisms like the sync object defined in /3/ for a time coordination of the related drive units.

The *interpolation data record* contains the interpolation data; the data type of the sub-indices of this structure manufacturer specific. Only the record size is fixed in the *size of data record* as sub-index of the *interpolation data configuration*

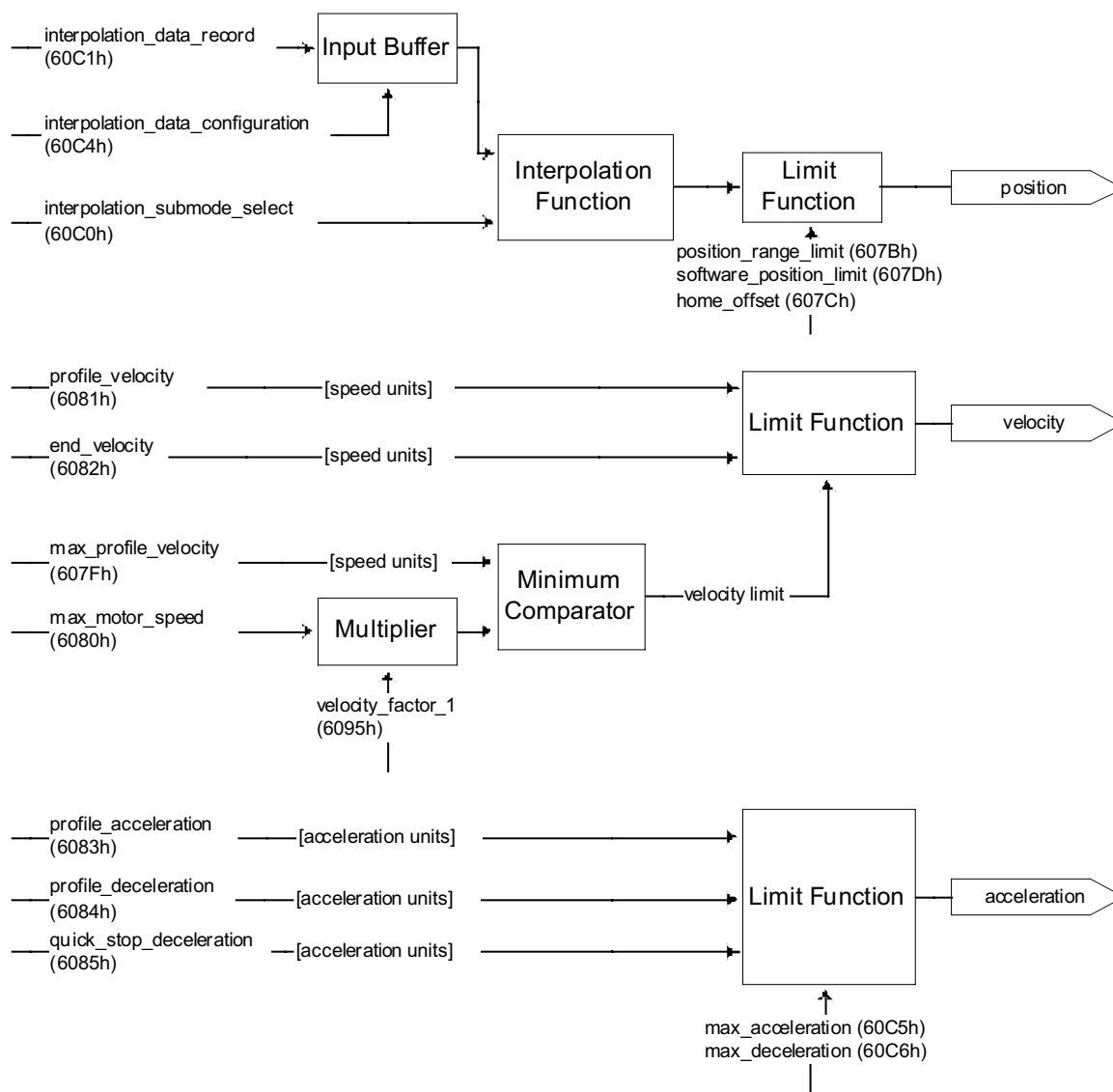
For synchronous operation the interpolation cycle time is defined by the object *interpolation time period*. For asynchronous operation the *interpolation time period* for each time slice must be included in the *interpolation data record*.

Time synchronization can be done by the Sync message (see /3/), a specific group sync signal (broadcast) or in specified time slices which are activated with the start signal.

The interpolated position mode allows a host controller to transmit a stream of interpolation data with either an implicit or explicit time reference to a drive unit. If the drive supports an input buffer, the interpolation data may be sent in bursts rather than continuously in real time. The actually available and the maximum size of the input buffer can be requested by a host using the *interpolation data configuration*. The buffer size is the number of *interpolation data records* which may be sent to a drive to fill the input buffer and it is not the size in bytes. Devices without input buffer capabilities have to accept at least one interpolation data item.

The interpolation algorithm is defined in the *interpolation sub mode select*. Linear interpolation is the default interpolation method. This requires only one interpolation data item to be buffered for the calculation of the next demand value. For each interpolation cycle, the drive will calculate a *position demand value* by interpolating positions over a period of time.

Optionally the common limit functions for speed, acceleration and deceleration may be applied to the interpolation data.



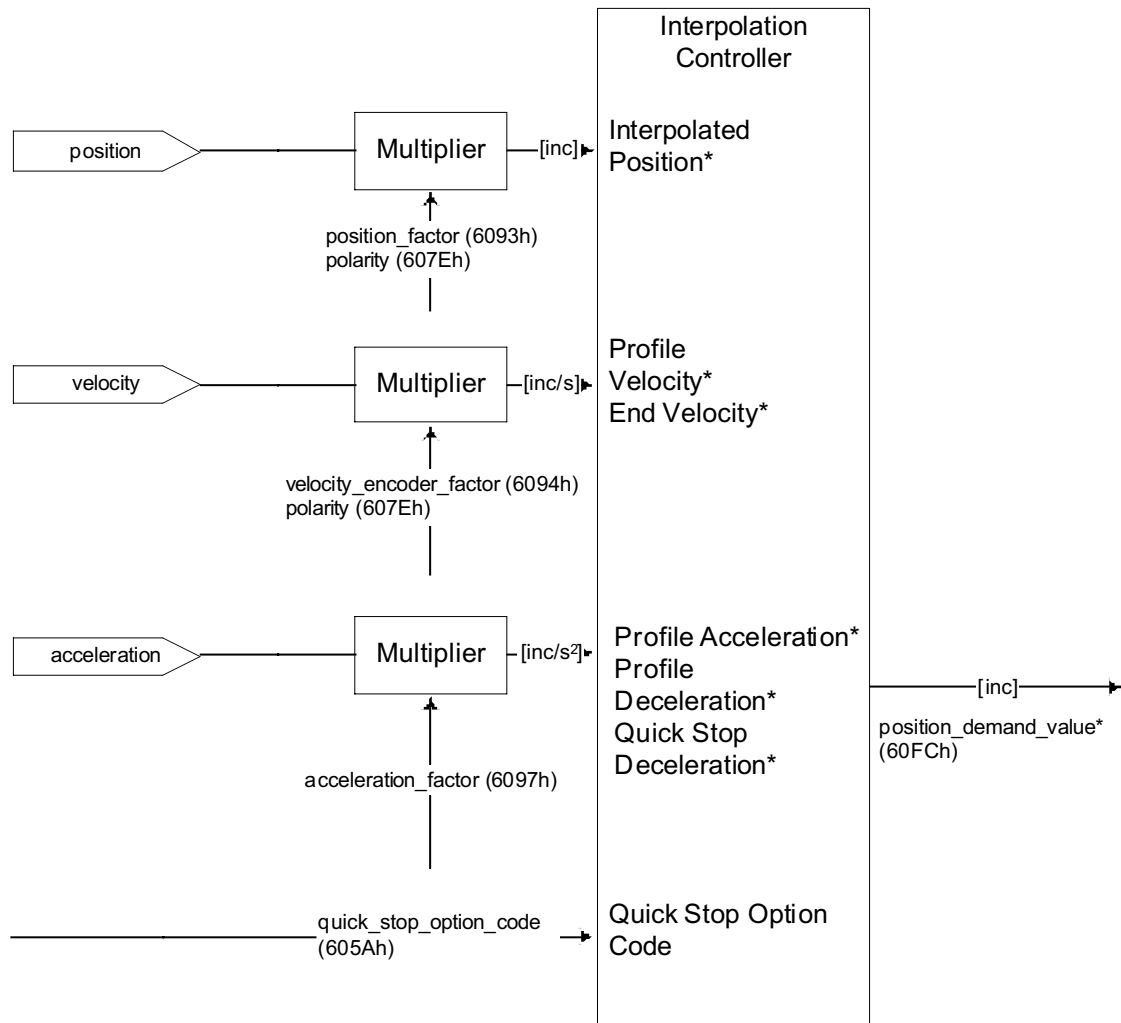


Figure 35: Interpolation controller

### 15.1.1 Input data description

Operating mode	Input parameters used
ip	<i>interpolation sub mode select, max profile velocity, profile acceleration, profile deceleration, quick stop deceleration*, quick stop mode</i>

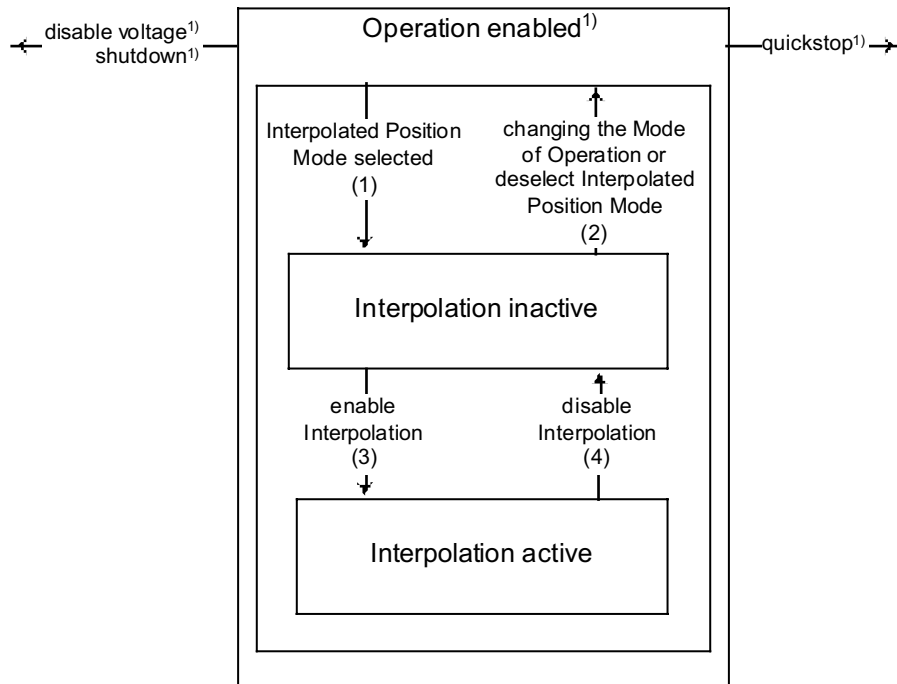
### 15.1.2 Output data description

The output values provided by the Interpolated position mode depend on the number and type of interpolation functions implemented by a manufacturer. For the predefined linear time interpolation the output is a *position demand value\**.

Operation mode	Output parameter used
ip	<i>position demand value*</i>

### 15.1.3 Internal states

#### 15.1.3.1 State description



<sup>1)</sup>see state machine

Figure 36: Internal states for the Interpolated position mode

- **Interpolation inactive**  
This state is entered when the device is in state OPERATION ENABLE and the Interpolated position mode is selected. The drive unit will accept input data and will buffer it for interpolation calculations, but it does not move the axes.
- **Interpolation active**  
This state is entered when the device is in state OPERATION ENABLE, the Interpolated position mode is selected and enabled. The drive unit will accept input data and it moves the axes.

#### 15.1.3.2 State transitions of the internal states

- State Transition 1: NO IP-MODE SELECTED => IP-MODE INACTIVE  
Event: Enter in the state OPERATION ENABLE with *controlword* and select ip mode with *modes of operation*
- State Transition 2: IP-MODE INACTIVE => NO IP-MODE SELECTED  
Event: Leave the state OPERATION ENABLE with *controlword* or select any other mode with *modes of operation* if it is allowed inside the state OPERATION ENABLE
- State Transition 3: IP-MODE INACTIVE => IP-MODE ACTIVE  
Event: Set bit *enable ip mode* (bit4) of the *controlword* while in ip mode and OPERATION ENABLE
- State Transition 4: IP-MODE ACTIVE => IP-MODE INACTIVE  
Event: Reset bit *enable ip mode* (bit4) of the *controlword* while in ip mode and OPERATION ENABLE

**15.1.3.3 Controlword of interpolated position mode**

15	9	8	7	6	5	4	3	0
(see 10.3.1)	Halt	(see 10.3.1)	reserved	Enable ip mode	(see 10.3.1)			
MSB					LSB			

Name	Value	Description
Enable ip mode	0	Interpolated position mode inactive
	1	Interpolated position mode active
Halt	0	Execute the instruction of bit 4
	1	Stop axle

Table 13: Interpolated position mode bits of the *controlword***15.1.3.4 Statusword of interpolated position mode**

15	14	13	12	11	10	9	0
(see 10.3.2)	reserved	ip mode active	(see 10.3.2)	Target reached	(see 10.3.2)		
MSB					LSB		

Name	Value	Description
Target reached	0	Halt = 0: Position not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: Position reached Halt = 1: Axle has velocity 0
ip mode active	0	Interpolated position mode inactive
	1	Interpolated position mode active

Table 14: Interpolated position mode bits of the *statusword***15.2 Complex data types****15.2.1 Interpolation time period record**

Index	Sub	Field in interpolation time period record	Data type
0080 <sub>h</sub>	0	number of entries	UNSIGNED8
	1	Interpolation time units	UNSIGNED8
	2	Interpolation time index	INTEGER8

**15.2.2 Interpolation data configuration record**

Index	Sub	Field in interpolation time period record	Data type
0081 <sub>h</sub>	0	number of entries	UNSIGNED8
	1	Maximum buffer size	UNSIGNED32
	2	Actual buffer size	UNSIGNED32
	3	Buffer organization	UNSIGNED8
	4	Buffer position	UNSIGNED16
	5	Size of data record	UNSIGNED8
	6	Buffer clear	UNSIGNED8

**15.3 Object dictionary entries****15.3.1 Objects defined in this chapter**

Index	Object	Name	Type	Attr.	M/O
60C0 <sub>h</sub>	VAR	Interpolation sub mode select	INTEGER16	rw	O
60C1 <sub>h</sub>	ARRAY	Interpolation data record	INTEGER32	rw	O
60C2 <sub>h</sub>	RECORD	Interpolation time period	Interpolation time period record	rw	O
60C3 <sub>h</sub>	ARRAY	Interpolation sync definition	UNSIGNED8	rw	O
60C4 <sub>h</sub>	RECORD	Interpolation data configuration	Interpolation data configuration record	rw	O

**15.3.2 Objects defined in other chapters**

Index	Object	Name	Type	Chapter
6040 <sub>h</sub>	VAR	Controlword	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	Statusword	UNSIGNED16	dc
605A <sub>h</sub>	VAR	Quick stop option mode	INTEGER16	dc
6060 <sub>h</sub>	VAR	Modes of operation	INTEGER8	dc
6061 <sub>h</sub>	VAR	Modes of operation display	INTEGER8	dc
6062 <sub>h</sub>	VAR	Position demand value	INTEGER32	pc
6063 <sub>h</sub>	VAR	Position actual value*	INTEGER32	pc
606A <sub>h</sub>	VAR	Sensor selection code	UNSIGNED8	pv
607F <sub>h</sub>	VAR	Max profile velocity	UNSIGNED32	pp
6089 <sub>h</sub>	VAR	Position notation index	INTEGER8	fg
608A <sub>h</sub>	VAR	Position dimension index	UNSIGNED8	fg
608B <sub>h</sub>	VAR	Velocity notation index	INTEGER8	fg
608C <sub>h</sub>	VAR	Velocity dimension index	UNSIGNED8	fg
608D <sub>h</sub>	VAR	Acceleration notation index	INTEGER8	fg
608E <sub>h</sub>	VAR	Acceleration dimension index	UNSIGNED8	fg
608F <sub>h</sub>	ARRAY	Position encoder resolution	UNSIGNED32	fg
6090 <sub>h</sub>	ARRAY	Velocity encoder resolution	UNSIGNED32	fg

Index	Object	Name	Type	Chapter
6091 <sub>h</sub>	ARRAY	Gear ratio	UNSIGNED32	fg
6092 <sub>h</sub>	ARRAY	Feed constant	UNSIGNED32	fg
6093 <sub>h</sub>	ARRAY	Position factor	UNSIGNED32	fg
6094 <sub>h</sub>	ARRAY	Velocity encoder factor	UNSIGNED32	fg
6095 <sub>h</sub>	ARRAY	Velocity factor 1	UNSIGNED32	fg
6098 <sub>h</sub>	ARRAY	Velocity factor 2	UNSIGNED32	fg
6097 <sub>h</sub>	ARRAY	Acceleration factor	UNSIGNED32	fg
60C5 <sub>h</sub>	VAR	Max acceleration	INTEGER32	pp
60C6 <sub>h</sub>	VAR	Max deceleration	INTEGER32	pp

## 15.4 Object descriptions

### 15.4.1 Object 60C0<sub>h</sub>: *Interpolation sub mode select*

For the interpolated position mode a manufacturer may offer different interpolation algorithms. This object reflects or changes the actually chosen interpolation mode.

#### OBJECT DESCRIPTION

INDEX	60C0 <sub>h</sub>
Name	Interpolation sub mode select
Object Code	VAR
Data Type	INTEGER16
Category	Optional

#### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

#### DATA DESCRIPTION

Value	Description
-32768..-1	manufacturer specific
0	Linear interpolation
+1..+32767	reserved

### 15.4.2 Object 60C1<sub>h</sub>: *Interpolation data record*

The *interpolation data record* are the data words which are necessary to perform the interpolation algorithm. The number N of data words in the record is defined by *interpolation data configuration*. The interpretation of the data words in *interpolation data record* may vary with the different possible interpolation modes as set by the *interpolation sub mode select*.

For the linear interpolation mode each interpolation data record simply can be regarded as a new position set-point. To describe a cubic spline interpolation e.g., four or more data words are needed for the spline coefficients, and further interpolation parameters.

After the last item of an *interpolation data record* is written to the devices input buffer, the pointer of the buffer is automatically incremented to the next buffer position.

## OBJECT DESCRIPTION

INDEX	60C1 <sub>h</sub>
Name	Interpolation data record
Object Code	ARRAY
Data Type	60C0 <sub>h</sub> < 0 : manufacturer specific 60C0 <sub>h</sub> = 0 : INTEGER32 60C0 <sub>h</sub> > 0 : not defined
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 .. N
Default Value	No

Sub-Index	1
Description	x <sub>1</sub> the first parameter of ip function $f_{ip}(x_1, \dots, x_N)$
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	60C0 <sub>h</sub> < 0 : manufacturer specific 60C0 <sub>h</sub> = 0 : INTEGER32 60C0 <sub>h</sub> > 0 : not defined
Default Value	No

Sub-Index	2
Description	x <sub>2</sub> the second parameter of ip function $f_{ip}(x_1, \dots, x_N)$
Entry Category	Optional
Access	rw
PDO Mapping	Possible
Value Range	60C0 <sub>h</sub> < 0 : manufacturer specific 60C0 <sub>h</sub> = 0 : INTEGER32 60C0 <sub>h</sub> > 0 : not defined
Default Value	No

to



Sub-Index	N
Description	$x_N$ the N-th parameter of ip function $f_{ip}(x_1, \dots, x_N)$
Entry Category	Optional
Access	rw
PDO Mapping	Possible
Value Range	$60C0_h < 0$ : manufacturer specific $60C0_h = 0$ : INTEGER32 $60C0_h > 0$ : not defined
Default Value	No

#### 15.4.3 Object 60C2<sub>h</sub>: *Interpolation time period*

The *interpolation time period* is used for time synchronized interpolation position modes. The unit of the *interpolation time unit* is given in  $10^{\text{interpolation time index}}$  seconds.

##### OBJECT DESCRIPTION

INDEX	60C2 <sub>h</sub>
Name	Interpolation time period
Object Code	RECORD
Data Type	Interpolation time period record (0080 <sub>h</sub> )
Category	Optional

##### ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Interpolation time units
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	1

Sub-Index	2
Description	Interpolation time index
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	-128 .. 63
Default Value	-3

#### 15.4.4 Object 60C3<sub>n</sub>: Interpolation sync definition

Devices in the interpolation position mode often interact with other devices. Therefore it is necessary to define a communication object which is used to synchronize these interactions.

This can be done by the general Sync as described in /3/, or a specific group-sync-signal. Each reception of this trigger-signal or a specified number of occurrences of the trigger-signal can synchronize the devices; a second opportunity is to use fixed time slices for synchronization.

Value	Description
0	General Sync is used
1 .. 255	reserved

Table 15: Description of synchronize on group

#### OBJECT DESCRIPTION

INDEX	60C3 <sub>n</sub>
Name	Interpolation sync definition
Object Code	ARRAY
Data Type	UNSIGNED8
Category	Optional

#### ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Synchronize on group
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	0

Sub-Index	2
Description	ip sync every n event
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	1

#### 15.4.5 Object 60C4<sub>h</sub>: *Interpolation data configuration*

It is possible to offer different algorithms of interpolation. Most of them need a larger number of position to calculate the actual position the axes should reach. To enable the device to receive the needed data in advance a data space is used to store the positions and further data send by the host.

Value	Description
0	FIFO buffer
1	Ring buffer
2 .. 255	reserved

Table 16: Types of buffer organization

Value	Description
0	<ul style="list-style-type: none"> <li>Clear input buffer</li> <li>Access disabled</li> <li>Clear all ip data records</li> </ul>
1	Enable access to the input buffer for the drive functions
2 .. 255	reserved

Table 17: Description of buffer clear values

#### OBJECT DESCRIPTION

INDEX	60C4 <sub>h</sub>
Name	Interpolation data configuration
Object Code	RECORD
Data Type	Interpolation data configuration record (0081 <sub>h</sub> )
Category	Optional

#### ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	6
Default Value	6

Sub-Index	1
Description	Maximum buffer size
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	1

Sub-Index	2
Description	Actual buffer size
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

Sub-Index	3
Description	Buffer organization
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	0

Sub-Index	4
Description	Buffer position
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	0

Sub-Index	5
Description	Size of data record
Entry Category	Mandatory
Access	wo
PDO Mapping	Possible
Value Range	1 .. 254
Default Value	1

Sub-Index	6
Description	Buffer clear
Entry Category	Mandatory
Access	wo
PDO Mapping	Possible
Value Range	UNSIGNED8
Default Value	0

## 15.5 Functional description

### 15.5.1 Interpolated position mode

A drive can be controlled and supervised by the *controlword* and the *statusword* respectively. To choose the operation mode, the *modes of operation* is used. The activated operation mode is monitored by *modes of operation display*.

A drive manufacturer has to specify the way the device handles a just received interpolation data record. This can be in a way corresponding to the standard position mode, or might be a more complex algorithm. The standard method is to apply new data immediately, respectively after the next synchronization signal.

An input buffer for interpolation data records is not mandatory, although it eases the data exchange between a host and a drive unit. The real-time requirements to the CAN-bus as well as to the drive unit decrease in this case, because an input buffer decouples the data processing in the drive from the data transmission via the bus line.

### 15.5.2 Linear interpolated position mode with several axes

In order to follow a two- or more-dimensional curve through the space with a defined speed, a host (an interpolation controller or a PLC) calculates the different positions  $P_i$  for each set of coordinates which have to be reached at specified times  $t_i$ .

To use the interpolation mode with several axes the host calculates the next or more positions and timestamps, and transmits them to the different axes. For each set-point  $P_i$  the interpolation controller has to calculate  $x_i$ ,  $y_i$ ... and  $t_i$ . Each axle gets a set of *interpolation data records* which each axle has to process internally independent from the other axes according to the chosen interpolation mode.

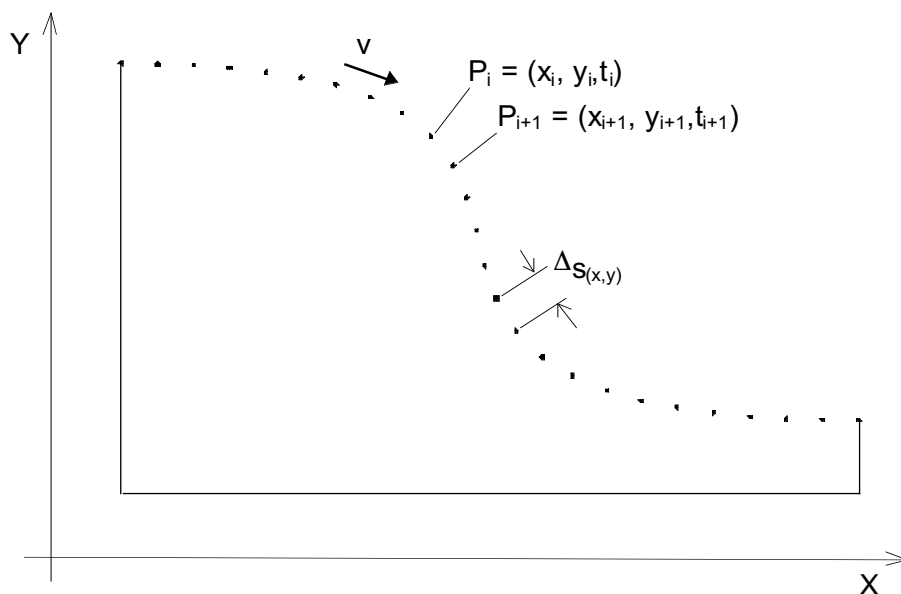


Figure 37: Interpolation for two axes

In a centralized drive system with a remote motion device doing the interpolation calculation, a central clocking scheme for synchronization of the different axes based on any kind of sync-signal is used. This results in a movement depending on the calculation cycle time of the interpolation controller. The velocity becomes more or less a fixed value for each axle.

calculated positions	ip data records for		
	x-axle	y-axle	z-axle
$P_i$	$x_i, t_i$	$y_i, t_i$	$z_i, t_i$
$P_{i+1}$	$x_{i+1}, t_{i+1}$	$y_{i+1}, t_{i+1}$	$z_{i+1}, t_{i+1}$
$P_{i+2}$	$x_{i+2}, t_{i+2}$	$y_{i+2}, t_{i+2}$	$z_{i+2}, t_{i+2}$
$P_{i+3}$	$x_{i+3}, t_{i+3}$	$y_{i+3}, t_{i+3}$	$z_{i+3}, t_{i+3}$
•	•	•	•
•	•	•	•
•	•	•	•
$P_{i+n}$	$x_{i+n}, t_{i+n}$	$y_{i+n}, t_{i+n}$	$z_{i+n}, t_{i+n}$

Table 18: Position calculation in interpolated position mode for several axes

In decentralized motion systems a host starts all relevant axes by changing the mode-internal state to interpolation active after preparing and sending one or more *interpolation data records* to all axes and synchronizes them by a (group) sync-signal. Each axle calculates internally and independently the necessary speed and acceleration needed to move from one position to the next. This can be done by calculating a linear or any other move between two given position set-points. Along this track every axle controls the movement between the set-points independently from the other axes. The axes may continue their move, as long as there is enough data to continue the calculations. Therefore it is easy to use the input buffer to give data records ahead.

With this information each axle can act like it is shown in Figure 37.

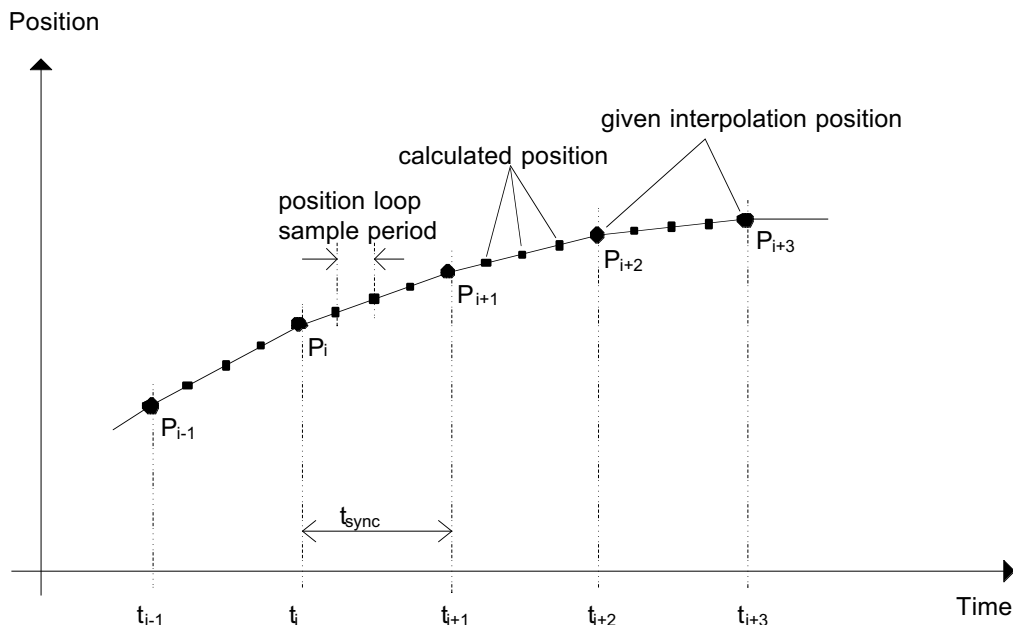


Figure 38: Linear interpolation for one axle

### 15.5.3 Buffer strategies for the interpolated position mode

If a device provides an input buffer for *interpolation data records* its size can be organized by a host using the *interpolation data configuration*. The host splits the available buffer capacity into pages which have the size of one *interpolation data record* each. This is done by *size of data record*. If one

page remains, which can not keep one complete data record, it can not be used. After the reorganization of the input buffer all previous stored data will be lost. All devices supporting the interpolated position mode need to implement an input buffer, which at least can keep one *interpolation data record*.

The content of the buffer items can only be accessed via the *interpolation data record*.

Commonly, first-in-first-out (FIFO) structures or ring buffers are used as input buffers.

- **FIFO:**  
If the buffer is organized as FIFO, every new received *interpolation data record* is placed at the end of the queue, and the device takes the next data record from the top of the queue. When the last item of a data record is stored, the buffer pointer is incremented in order to point to the next buffer position. For this buffer principle the object *buffer position* does not have any influence.
- **Ring buffer:**  
If the buffer is structured as a ring, the host can place an *interpolation data record* into any valid position in the ring by changing the pointer defined in *buffer position*. Without changing the *buffer position* all data records will be written at the same location. The drive reads the next entry out of the buffer by an internal ring pointer. It is set to the first data record with *buffer clear*, and after the reorganization of the input buffer.

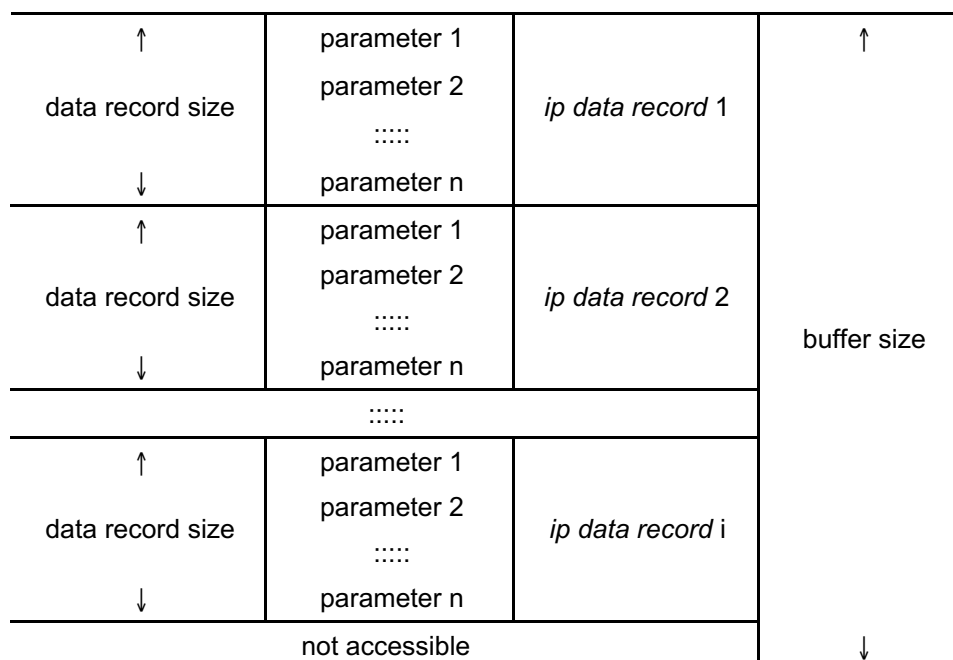


Figure 39: Input buffer organization

# 16 PROFILE VELOCITY MODE

## 16.1 General Information

The profile velocity mode includes the following sub-functions:

- demand value input via trajectory generator
- velocity capture using position sensor or velocity sensor
- velocity control function with appropriate input and output signals
- limitation of *torque demand value*
- monitoring of the *profile velocity* using a window-function
- monitoring of *velocity actual value* using a threshold

The operation of the reference value generator and its input parameters:

- profile velocity,
- profile acceleration,
- profile deceleration,
- emergency stop, and
- motion profile type

are described in the profile position mode (see 12).

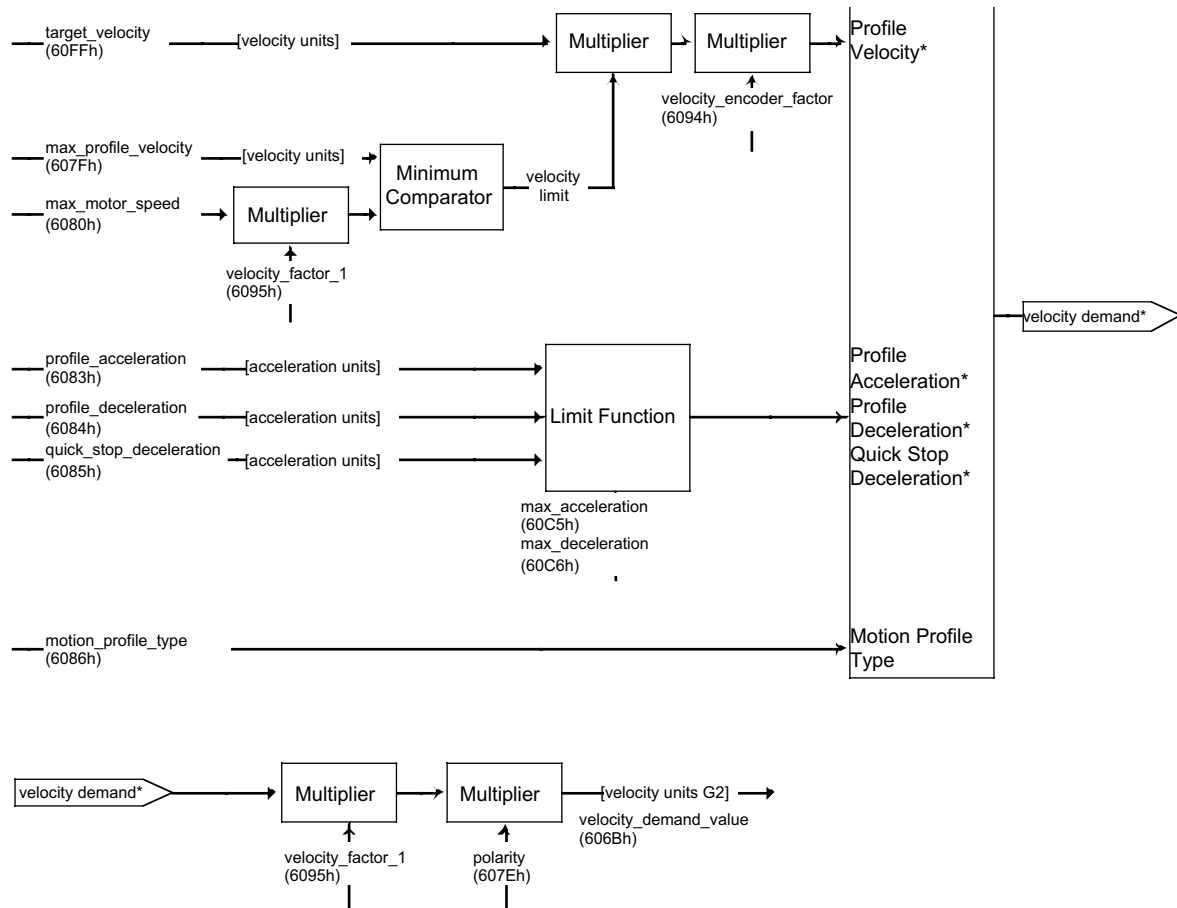
Various sensors can be used for velocity capture. In particular the aim is that costs should be reduced and the system should be simplified by evaluating position and velocity using a common sensor, such as is possible using a resolver or an encoder.

The velocity control function is not specified more precisely at this point as it is highly manufacturer specific, but the format and maximum number of control coefficients are established.

The velocity controller calculates a torque variable. This is added to a torque pre-control calculated by the trajectory generator and limited to a *torque max value*. The limited total is used as input to the torque controller as a *torque demand value*.

Monitoring functions for the *velocity actual value* provide status information for super-ordinated systems.





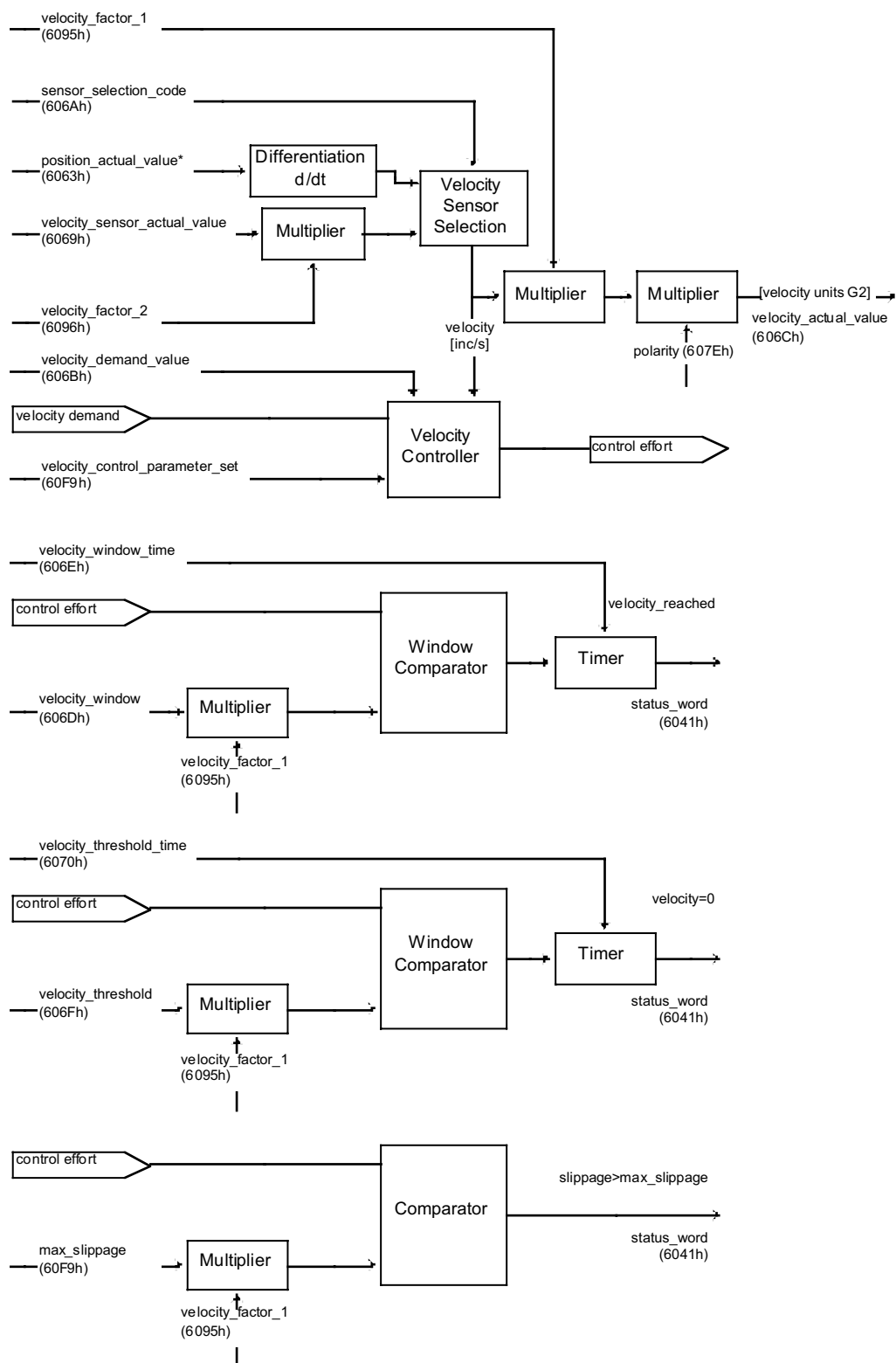


Figure 40: Structure of the profile velocity mode

**16.1.1 Input data description**

Operating mode	Input parameters used
pv	<i>target velocity, velocity factor 1, velocity factor 2, velocity window, velocity window time, velocity threshold, velocity threshold time, max slippage, profile acceleration, profile deceleration, quick stop deceleration, max acceleration, max deceleration, polarity, quick stop option code, motion profile type, max profile velocity, max motor speed</i>

**16.1.2 Output data description**

Operation mode	Output parameter used
pv	<i>velocity actual value, velocity demand value, statusword</i>

**16.1.3 Internal states**

The profile velocity mode has to be controlled by the bits of the *controlword* and *statusword*.

**16.1.3.1 Controlword of profile velocity mode**

15	9	8	7	6	4	3	0
(see 10.3.1)	Halt	(see 10.3.1)	reserved			(see 10.3.1)	
MSB				LSB			

Name	Value	Description
Halt	0	Execute the motion
	1	Stop axle

Table 19: Profile velocity mode bits of the *controlword*

**16.1.3.2 Statusword of profile velocity mode**

15	14	13	12	11	10	9	0
(see 10.3.2)	Max slippage error		Speed	(see 10.3.2)	Target reached	(see 10.3.2)	
MSB				LSB			

Name	Value	Description
Target reached	0	Halt = 0: <i>Target velocity</i> not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: <i>Target velocity</i> reached Halt = 1: Axle has velocity 0
Speed	0	Speed is not equal 0
	1	Speed is equal 0
Max slippage error	0	Maximum slippage not reached
	1	Maximum slippage reached

Table 20: Profile velocity mode bits of the *statusword*

## 16.2 Object dictionary entries

### 16.2.1 Objects defined in this chapter

Index	Object	Name	Type	Attr.	M/O
6069 <sub>h</sub>	VAR	Velocity sensor actual value	INTEGER32	ro	M
606A <sub>h</sub>	VAR	Sensor selection code	INTEGER16	rw	O
606B <sub>h</sub>	VAR	Velocity demand value	INTEGER32	ro	M
606C <sub>h</sub>	VAR	Velocity actual value	INTEGER32	ro	M
606D <sub>h</sub>	VAR	Velocity window	UNSIGNED16	rw	O
606E <sub>h</sub>	VAR	Velocity window time	UNSIGNED16	rw	O
606F <sub>h</sub>	VAR	Velocity threshold	UNSIGNED16	rw	O
6070 <sub>h</sub>	VAR	Velocity threshold time	UNSIGNED16	rw	O
60FF <sub>h</sub>	VAR	Target velocity	INTEGER32	rw	M
60F8 <sub>h</sub>	VAR	Max slippage	INTEGER32	rw	O
60F9 <sub>h</sub>	ARRAY	Velocity control parameter set	UNSIGNED16	rw	O

### 16.2.2 Objects defined in other chapters

Index	Object	Name	Type	Chapter
6040 <sub>h</sub>	VAR	Controlword	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	Statusword	UNSIGNED16	dc
6063 <sub>h</sub>	VAR	Position actual value*	INTEGER32	pc
6069 <sub>h</sub>	VAR	Velocity sensor actual value	INTEGER32	pv
6071 <sub>h</sub>	VAR	Target torque	INTEGER16	tq
6072 <sub>h</sub>	VAR	Max torque	UNSIGNED16	tq
607E <sub>h</sub>	VAR	Polarity	UNSIGNED8	fg
607F <sub>h</sub>	VAR	Max profile velocity	UNSIGNED32	pp
6080 <sub>h</sub>	VAR	Max motor speed	UNSIGNED32	pp
6083 <sub>h</sub>	VAR	Profile acceleration	UNSIGNED32	pp
6084 <sub>h</sub>	VAR	Profile deceleration	UNSIGNED32	pp
6085 <sub>h</sub>	VAR	Quick stop deceleration	UNSIGNED32	pp
6086 <sub>h</sub>	VAR	Motion profile type	INTEGER16	pp
6094 <sub>h</sub>	ARRAY	Velocity encoder factor	UNSIGNED32	fg
6095 <sub>h</sub>	ARRAY	Velocity factor 1	UNSIGNED32	fg
6096 <sub>h</sub>	ARRAY	Velocity factor 2	UNSIGNED32	fg

## 16.3 Object description

The factors necessary for scaling

- *velocity reference factor*,
- *velocity factor 1*, and
- *velocity factor 2*

have a linear relationship and therefore they are described in the factor group. The *polarity* is described in the factor group as well.

**16.3.1 Object 6069<sub>h</sub>: Velocity sensor actual value**

The *velocity sensor actual value* describes the value read from a velocity encoder (if present) in increments (in the case of encoders) and in increments per second (in the case of tachometers and AD converters). This value is scaled to the format of the position encoder using the scaling factor *velocity factor 2*.

**OBJECT DESCRIPTION**

INDEX	6069 <sub>h</sub>
Name	Velocity sensor actual value
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pv supported

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**16.3.2 Object 606A<sub>h</sub>: Sensor selection code**

The source of the *velocity sensor actual value* can be determined using the *sensor selection code*. This determines whether a differentiated position signal or the signal from a separate velocity sensor has to be evaluated.

**OBJECT DESCRIPTION**

INDEX	606A <sub>h</sub>
Name	Sensor selection code
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**DATA DESCRIPTION**

Value	Description
0000 <sub>h</sub>	Actual velocity value from position encoder
0001 <sub>h</sub>	Actual velocity value from velocity encoder
0002 <sub>h</sub> .. 7FFF <sub>h</sub>	reserved
8000 <sub>h</sub> ... FFFF <sub>h</sub>	manufacturer specific

**16.3.3 Object 606B<sub>n</sub>: *Velocity demand value***

The output value of the trajectory generator may be corrected by the output value of the position control function. It is then provided as a demand value for the velocity controller and given in the velocity units.

**OBJECT DESCRIPTION**

INDEX	606B <sub>n</sub>
Name	Velocity demand value
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pv supported

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**16.3.4 Object 606C<sub>n</sub>: *Velocity actual value***

The *velocity actual value* is also represented in velocity units and is coupled to the velocity used as input to the velocity controller.

**OBJECT DESCRIPTION**

INDEX	606C <sub>n</sub>
Name	Velocity actual value
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pv supported

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**16.3.5 Object 606D<sub>n</sub>: *Velocity window***

The *velocity window* monitors whether the required process velocity has been achieved after an eventual acceleration or deceleration (braking) phase. It is given in velocity units.

**OBJECT DESCRIPTION**

INDEX	606D <sub>n</sub>
Name	Velocity window
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**16.3.6 Object 606E<sub>h</sub>: Velocity window time**

The corresponding bit 10 *target reached* is set in the *statusword* when the difference between the *target velocity* and the *velocity actual value* is within the *velocity window* longer than the *velocity window time*. The value of the velocity window time is given in multiples of milliseconds.

## OBJECT DESCRIPTION

INDEX	606E <sub>h</sub>
Name	Velocity window time
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	0

**16.3.7 Object 606F<sub>h</sub>: Velocity threshold**

As soon as the *velocity actual value* exceeds the *velocity threshold* longer than the *velocity threshold time* bit 12 *velocity = 0* is reset in the *statusword*. Below this threshold the bit is set and indicates that the axle is stationary. The value is given in velocity units.

## OBJECT DESCRIPTION

INDEX	606F <sub>h</sub>
Name	Velocity threshold
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**16.3.8 Object 6070<sub>h</sub>: Velocity threshold time**

The velocity threshold time is given in multiples of milliseconds (for description see 16.3.7).

**OBJECT DESCRIPTION**

INDEX	6070 <sub>h</sub>
Name	Velocity threshold time
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**16.3.9 Object 60FF<sub>h</sub>: Target velocity**

The *target velocity* is the input for the trajectory generator and the value is given in velocity units.

**OBJECT DESCRIPTION**

INDEX	60FF <sub>h</sub>
Name	Target velocity
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pv supported

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**16.3.10 Object 60F8<sub>h</sub>: Max slippage**

The *max slippage* monitors whether the maximal slippage has actually been reached.

This value is scaled to the format of the position encoder using the scaling factor *velocity factor 2* and given in velocity units.

When the *max slippage* has been reached, the corresponding bit 13 *max slippage error* in the *statusword* will be set. The reaction of the drive when the *max slippage* error occurs, is manufacturer specific.



## OBJECT DESCRIPTION

INDEX	60F8 <sub>h</sub>
Name	Max slippage
Object Code	VAR
Data Type	INTEGER32
Category	Conditional; Mandatory, if pv supported

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	No

**16.3.11 Object 60F9<sub>h</sub>: Velocity control parameter set**

In order to control the behavior of the velocity control loop, one or more parameters are necessary. This object defines a rudimentary set of three parameters for a PID-control which may be enlarged by the manufacturer up to 255 parameters.

## OBJECT DESCRIPTION

INDEX	60F9 <sub>h</sub>
Name	Velocity control parameter set
Object Code	ARRAY
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2 .. 254
Default Value	No

Sub-Index	1
Description	Gain
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

Sub-Index	2
Description	$T_I$ – integration time constant
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

Sub-Index	3
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	UNSIGNED16
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	UNSIGNED16
Default Value	No

#### 16.4 Functional description

The actual velocity can be obtained through differentiation from the position encoder and is represented in position encoder increments.

# 17 PROFILE TORQUE MODE

## 17.1 General information

This chapter describes the profile torque mode. The profile torque mode allows a host (external) control system (i.e. closed-loop speed controller, open-loop transmission force controller) to transmit the *target torque* value, which is processed via the trajectory generator. The *torque slope* and *torque profile type* parameters are required.

Should the host control system switch the *controlword* bit 8 (halt) from 0 to 1 or from 1 to 0, than the trajectory generator ramps its *control effort* output down to zero, respectively up to the *target torque*. In both cases the trajectory generator takes the *torque slope* and *torque profile type* into consideration.

All definitions within this document refer to rotating motors. Using linear motors instead requires that all "torque" objects refer to a "force" instead. For the sake of simplicity, the objects are not duplicated and their names should not be modified. As an example, the linear motor target force must be transmitted using the *target torque* object. Refer to the object descriptions for additional information.

The manufacturer-specific torque control and power-stage functions are not described as they fall beyond the scope of this standard. They are only mentioned for showing how some parameters affect them. As an example the closed-loop torque control coefficients (if any) are to be defined and described by the manufacturer.

The *torque control parameters*, *power stage parameters* and *motor parameters* are defined as objects in order that they can be handled (i.e. downloaded) in a standard way. Their detailed data content is manufacturer-specific.

The *torque demand*, *torque actual value*, *current actual value* and *DC link voltage* may be available to the user as parameters, if they are monitored.

Depending on the drive and motor technologies the manufacturer-specific torque control function has to be active when another mode is selected (hm, pv, pc, ip). In such a case, selecting one of these modes implicitly activates the torque control and power-stage function, using the *control effort* as input.

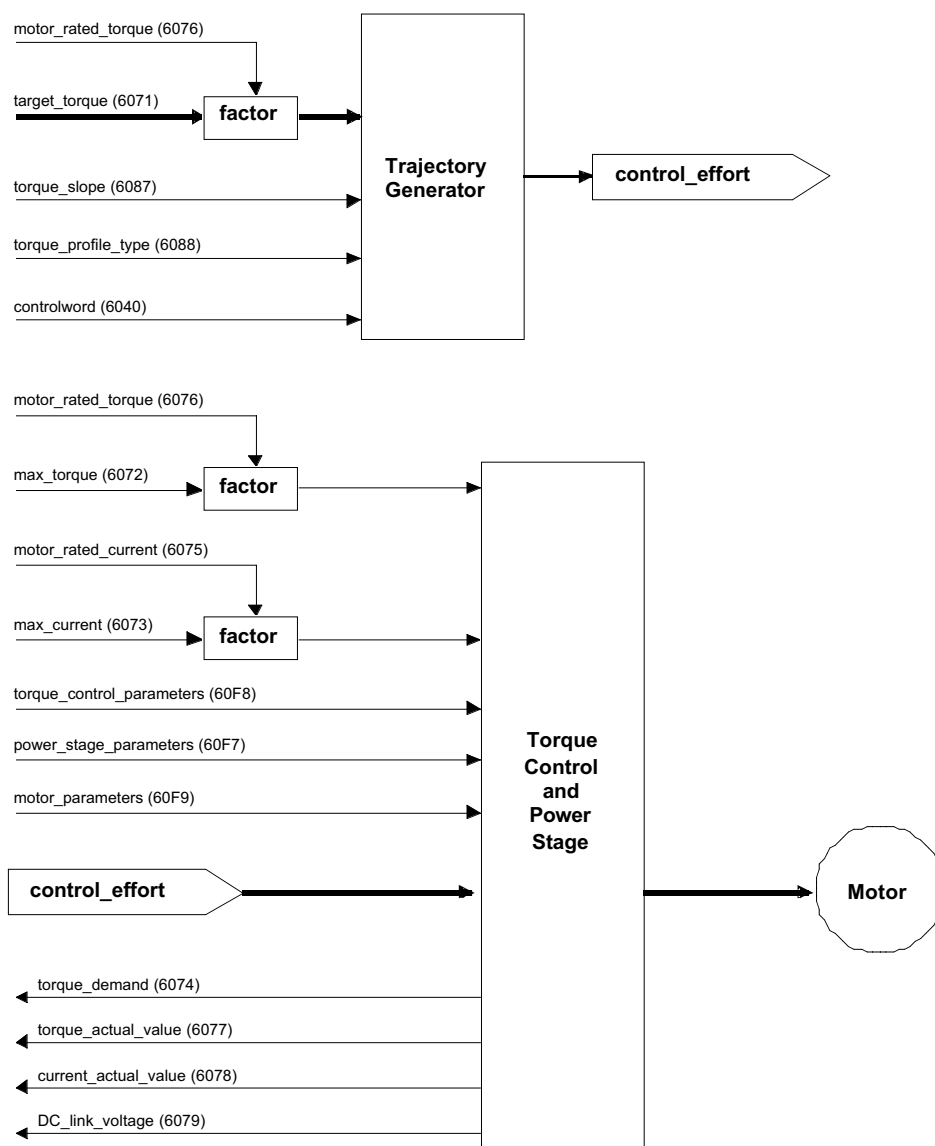
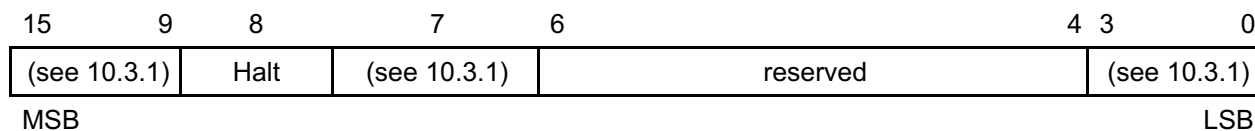


Figure 41: Structure of the profile torque mode

### 17.1.1 Internal states

#### 17.1.1.1 Controlword of profile torque mode



Name	Value	Description
Halt	0	Execute motion
	1	Stop axle

Table 21: Profile torque mode bits of the *controlword*

**17.1.1.2 Statusword of profile torque mode**

15	14	13	12	11	10	9	0		
(see 10.3.2)		reserved		(see 10.3.2)		Target reached		(see 10.3.2)	
MSB						LSB			

Name	Value	Description
Target reached	0	Target torque not reached
	1	Target torque reached

Table 22: Profile torque mode bits of the *statusword***17.2 Object dictionary entries****17.2.1 Objects defined in this chapter**

Index	Object	Name	Type	Attr.	M/O
6071 <sub>h</sub>	VAR	Target torque	INTEGER16	rw	M
6072 <sub>h</sub>	VAR	Max torque	UNSIGNED16	rw	O
6073 <sub>h</sub>	VAR	Max current	UNSIGNED16	rw	O
6074 <sub>h</sub>	VAR	Torque demand value	INTEGER16	ro	O
6075 <sub>h</sub>	VAR	Motor rated current	UNSIGNED32	rw	O
6076 <sub>h</sub>	VAR	Motor rated torque	UNSIGNED32	rw	O
6077 <sub>h</sub>	VAR	Torque actual value	INTEGER16	ro	O
6078 <sub>h</sub>	VAR	Current actual value	INTEGER16	ro	O
6079 <sub>h</sub>	VAR	DC link circuit voltage	UNSIGNED32	ro	O
6087 <sub>h</sub>	VAR	Torque slope	UNSIGNED32	rw	M
6088 <sub>h</sub>	VAR	Torque profile type	INTEGER16	rw	M
60F7 <sub>h</sub>	RECORD	Power stage parameters	(manufacturer specific)	rw	O
60F6 <sub>h</sub>	RECORD	Torque control parameters	(manufacturer specific)	rw	O

**17.2.2 Objects defined in other chapters**

Index	Object	Name	Type	Chapter
6040 <sub>h</sub>	VAR	Controlword	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	Statusword	UNSIGNED16	dc
6410 <sub>h</sub>	RECORD	Motor data	(manufacturer specific)	ce

### 17.3 Object description

#### 17.3.1 Object 6071<sub>h</sub>: *Target torque*

This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand of rated torque.

##### OBJECT DESCRIPTION

INDEX	6071 <sub>h</sub>
Name	Target torque
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if tq supported

##### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

#### 17.3.2 Object 6072<sub>h</sub>: *Max torque*

This value represents the maximum permissible torque in the motor and is given per thousand of rated torque.

##### OBJECT DESCRIPTION

INDEX	6072 <sub>h</sub>
Name	Max torque
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

##### ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	0

#### 17.3.3 Object 6073<sub>h</sub>: *Max current*

This value represents the maximum permissible torque creating current in the motor and is given per thousand of rated current.

##### OBJECT DESCRIPTION

INDEX	6073 <sub>h</sub>
Name	Max current
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	0

**17.3.4 Object 6074<sub>h</sub>: Torque demand value**

This parameter is the output value of the torque limit function (if the torque control and power-stage function are available). The value is given per thousand of rated torque.

## OBJECT DESCRIPTION

INDEX	6074 <sub>h</sub>
Name	Torque demand value
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**17.3.5 Object 6075<sub>h</sub>: Motor rated current**

This value is taken from the motor name plate and is entered as multiples of milliamp. Depending on the motor and drive technology this current may be either DC, peak or rms (root-mean-square) current. All relative current data refers to this value.

## OBJECT DESCRIPTION

INDEX	6075 <sub>h</sub>
Name	Motor rated current
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

**17.3.6 Object 6076<sub>h</sub>: Motor rated torque**

This value is taken from the motor name plate and is entered as multiples of mNm (mill Newtonmeter). All relative torque data refer to this value.

For linear motors, the object name is not changed, but the motor rated force value must be entered as multiples of mN (mill Newton).

**OBJECT DESCRIPTION**

INDEX	6076 <sub>h</sub>
Name	Motor rated torque
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

**17.3.7 Object 6077<sub>h</sub>: Torque actual value**

The *torque actual value* corresponds to the instantaneous torque in the drive motor. The value is given per thousand of rated torque.

**OBJECT DESCRIPTION**

INDEX	6077 <sub>h</sub>
Name	Torque actual value
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**17.3.8 Object 6078<sub>h</sub>: Current actual value**

The *current actual value* refers to the instantaneous current in the drive motor. The value is given per thousand of rated current.

**OBJECT DESCRIPTION**

INDEX	6078 <sub>h</sub>
Name	Current actual value
Object Code	VAR
Data Type	INTEGER16
Category	Optional



## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**17.3.9 Object 6079<sub>h</sub>: DC link circuit voltage**

This parameter describes the instantaneous DC link current voltage at the drive controller. The value is given in multiples of mill volts.

## OBJECT DESCRIPTION

INDEX	6079 <sub>h</sub>
Name	DC link circuit voltage
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

**17.3.10 Object 6087<sub>h</sub>: Torque slope**

This parameter describes the rate of change of torque in units of per thousand of rated torque per second.

## OBJECT DESCRIPTION

INDEX	6087 <sub>h</sub>
Name	Torque slope
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional; Mandatory, if tq supported

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	0

**17.3.11 Object 6088<sub>h</sub>: Torque profile type**

The *torque profile type* is used to select the type of torque profile used to perform a torque change.

**OBJECT DESCRIPTION**

INDEX	6088 <sub>h</sub>
Name	Torque profile type
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if tq supported

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**DATA DESCRIPTION**

Value	Description
0000 <sub>h</sub>	Linear ramp (trapezoidal profile)
0001 <sub>h</sub>	sin <sup>2</sup> ramp
0002 <sub>h</sub> .. 7FFF <sub>h</sub>	reserved
8000 <sub>h</sub> .. FFFF <sub>h</sub>	manufacturer specific

**17.3.12 Object 60F7<sub>h</sub>: Power stage parameters**

The *power stage parameters* object is used to handle (i.e. download) all manufacturer-specific power-stage parameters in a whole, in a standard way.

**OBJECT DESCRIPTION**

INDEX	60F7 <sub>h</sub>
Name	Power stage parameters
Object Code	RECORD
Data Type	manufacturer specific
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 .. 254
Default Value	No

Sub-Index	1
Description	manufacturer specific
Entry Category	Mandatory
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

Sub-Index	2
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

### 17.3.13 Object 60F6<sub>n</sub>: *Torque control parameters*

The *torque control parameters* object is used to handle (i.e. download) all manufacturer-specific torque control parameters in a whole, in a standard way.

#### OBJECT DESCRIPTION

INDEX	60F6 <sub>n</sub>
Name	Torque control parameters
Object Code	RECORD
Data Type	manufacturer specific
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 .. 254
Default Value	No

Sub-Index	1
Description	manufacturer specific
Entry Category	Mandatory
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

Sub-Index	2
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

to

Sub-Index	254
Description	manufacturer specific
Entry Category	Optional
Access	manufacturer specific
PDO Mapping	manufacturer specific
Value Range	manufacturer specific
Default Value	No

# 18 VELOCITY MODE

## 18.1 General description

The velocity mode is based on /5/ and /6/ and refers to the speed function group 1 of /6/.

The most frequently used devices with this mode are low-cost frequency inverters. But this profile could be used with all types of drives and other devices where it fits. Therefore data objects are almost 16 bit wide. The calculation of variables at the drive is possible by usual 8 bit microprocessors.

Most applications use a velocity set-point and a control word for switching the drive on and off.

Example for a minimal implementation of the velocity mode.

See chapter 10 for device control.

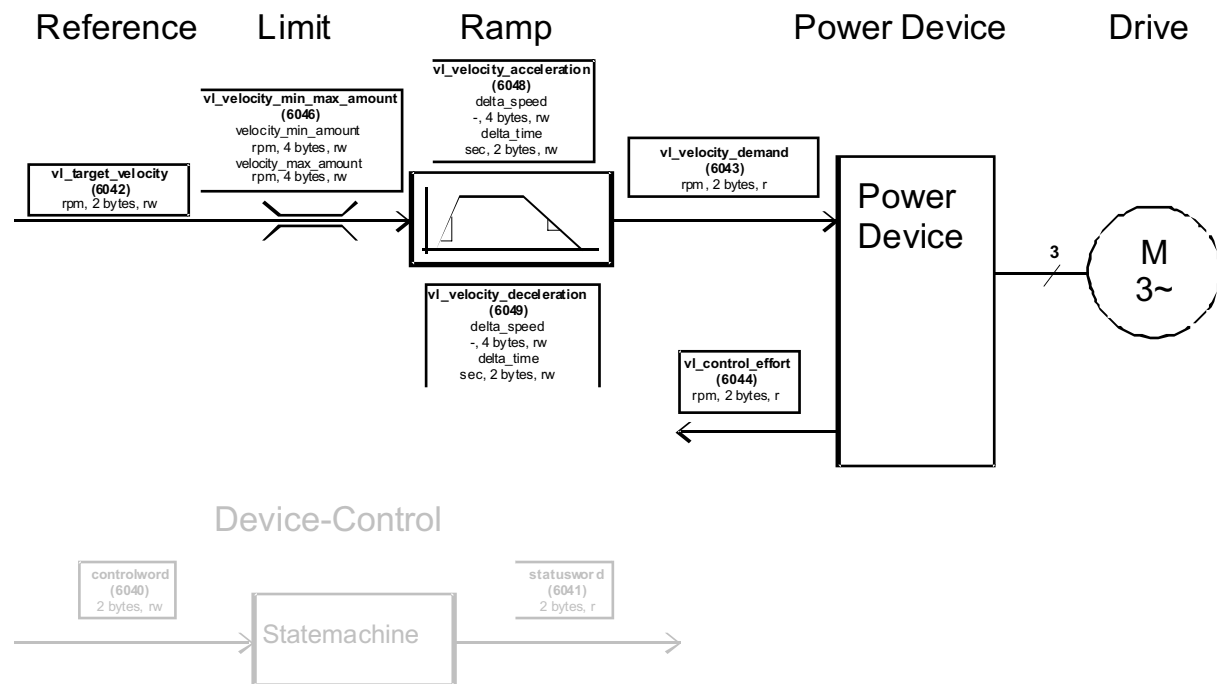


Figure 42: Example of a velocity mode application

### 18.1.1 Input data description

The velocity mode has the following input parameter:

Operating mode	Input parameters used
vl	vl target velocity, vl nominal percentage, vl pole number, vl dimension factor; vl velocity min max amount, vl velocity min max, vl velocity motor min max amount, vl velocity motor min max, vl frequency motor min max amount, vl frequency motor min max, vl velocity acceleration, vl velocity deceleration, vl velocity quick stop, vl ramp function time, vl slow down time, vl quick stop time, vl velocity reference, vl set-point factor

These objects are only used for the velocity mode.

### 18.1.2 Output data description

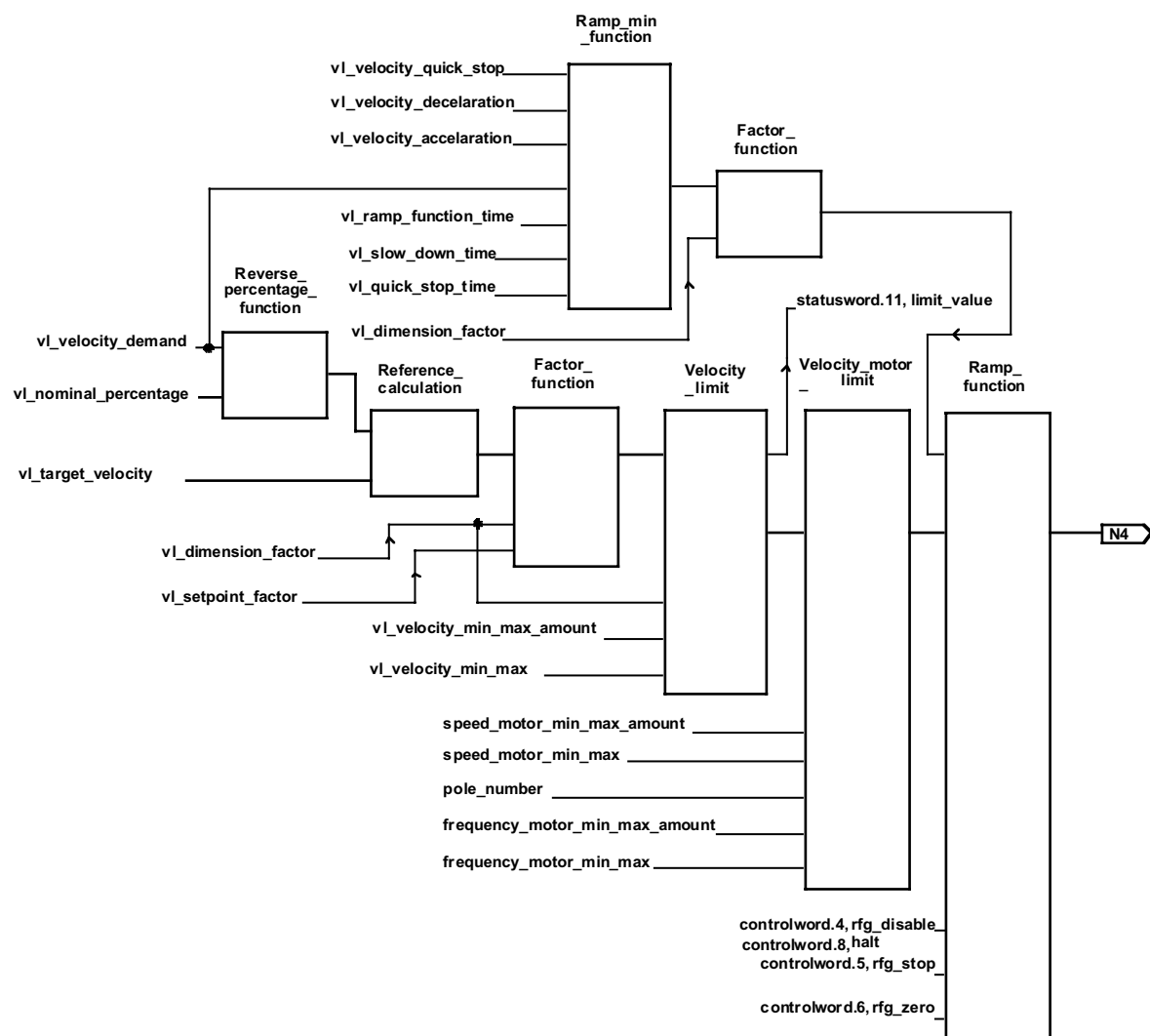
The velocity mode provides the following output parameters:

Operation mode	Output parameter used
vl	<i>vl control effort, vl manipulated velocity, vl percentage demand, vl actual percentage, vl velocity demand, vl manipulated percentage</i>

These objects are only used for the velocity mode.

### 18.1.3 Structure of the velocity mode

The diagram below shows the overall structure of the velocity mode. All mandatory and optional objects are used. It is not intended with it to specify implementations, but to describe the scope of functions. In these structures, the unit in which the velocity values in the speed functions are calculated is rpm. The descriptions of the drive functions refer to this structure. The device control is of course used in the velocity mode, but it is described in an extra chapter.



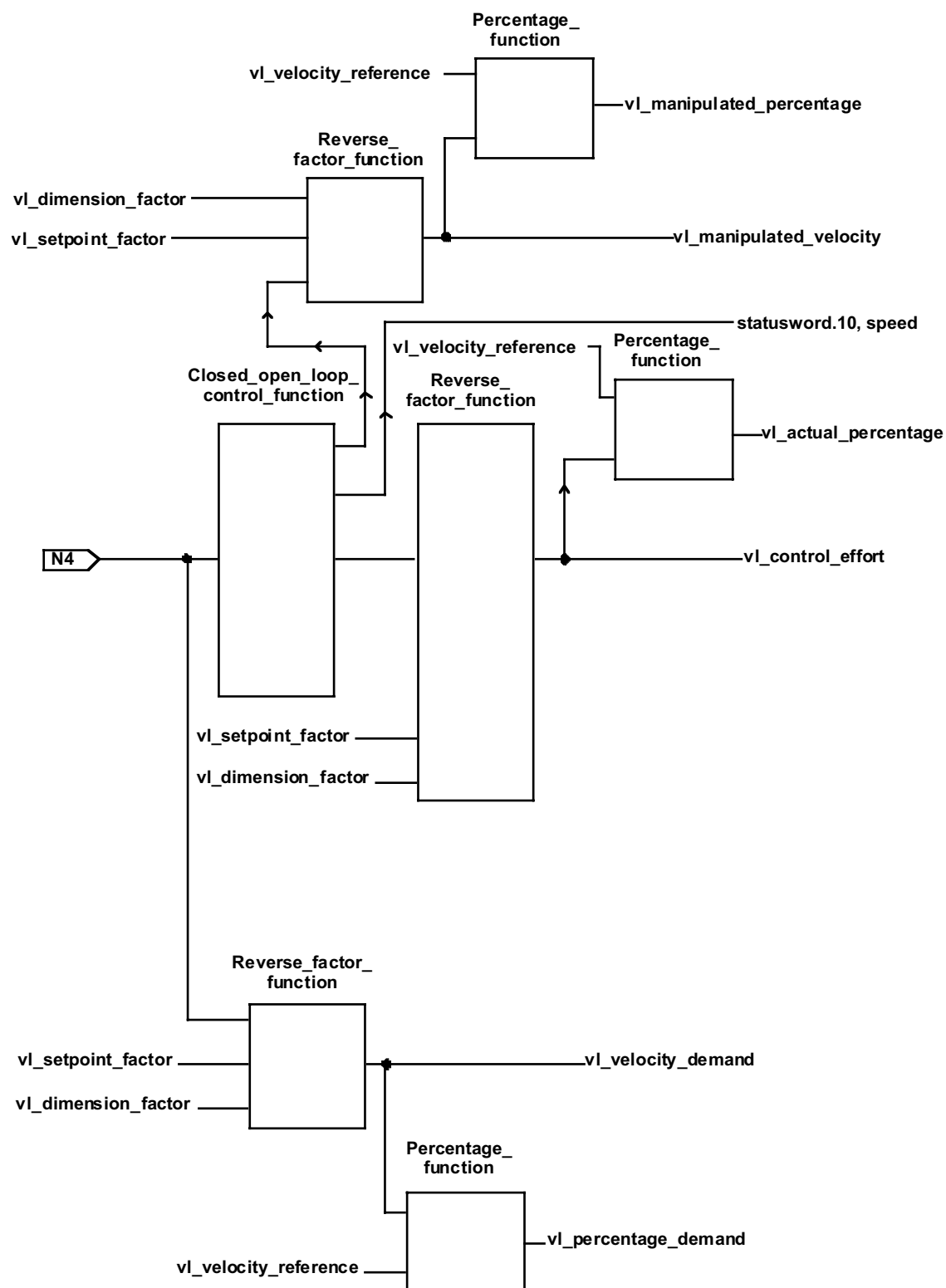


Figure 43: Velocity mode with all objects

All device using this profile and supporting the velocity mode have to implement the mandatory objects and there functionality. The diagram below shows the structure all devices will have at minimum.

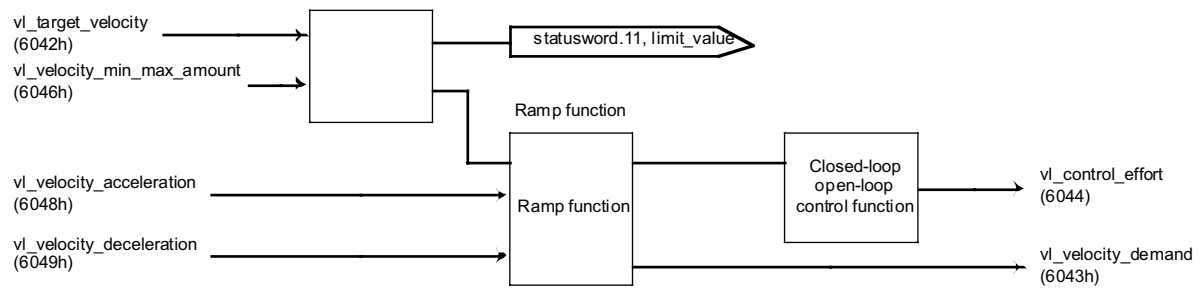


Figure 44: Velocity mode with mandatory objects only

### 18.1.4 Sub-function description

The velocity mode is composed of the following sub-functions:

- Reference calculation,
- Factor function, Reverse factor function,
- Percentage function, Reverse percentage function,
- Pole number function, Reverse pole number function,
- Velocity limit function,
- Velocity motor limit function,
- Ramp function,
- Ramp min function, and
- Closed open loop control function.

These sub-functions are only used for the velocity mode.

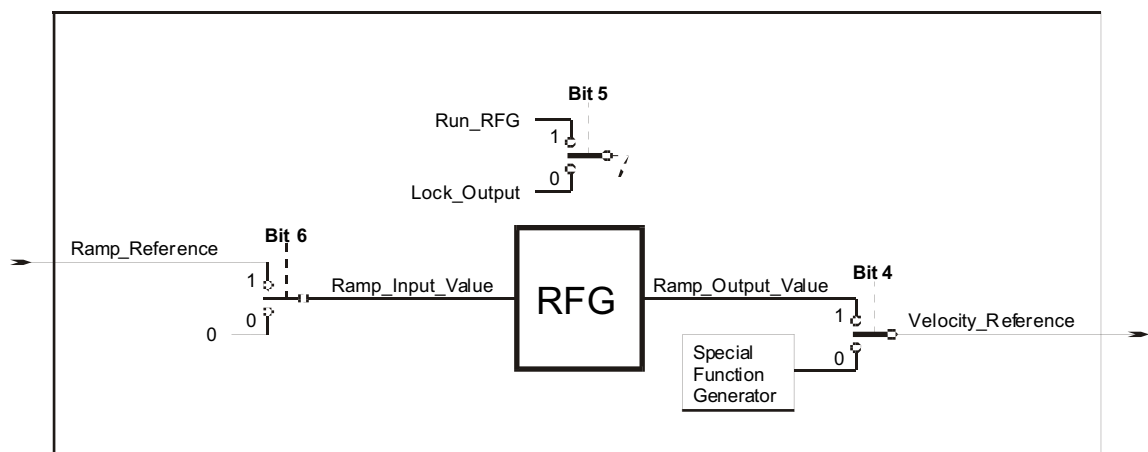
### 18.1.5 Internal states

#### 18.1.5.1 Controlword of velocity mode

15	9	8	7	6	5	4	3	0
(see 10.3.1)	Halt	(see 10.3.1)	rfg use ref	rfg unlock	rfg enable	(see 10.3.1)		
MSB							LSB	



Name	Value	Description
rfg enable	0	Velocity reference value is controlled in any other (manufacturer specific) way, e.g. by a test function generator or manufacturer specific halt function.
	1	Velocity reference value accords to ramp output value.
rfg unlock	0	Ramp output value is locked to current output value.
	1	Ramp output value follows ramp input value.
rfg use ref	0	Ramp input value is set to zero.
	1	Ramp input value accords to ramp reference.
Halt	0	Execute motion
	1	Stop axle

Table 23: Velocity mode bits of the *controlword*Figure 45: *Controlword* bits in the velocity mode

#### 18.1.5.2 Statusword of velocity mode

15	14	13	12	11	10	9	0		
(see 10.3.2)		reserved		(see 10.3.2)		reserved		(see 10.3.2)	
MSB					LSB				

## 18.2 Complex data types

### 18.2.1 vl velocity acceleration deceleration record

Index	Sub-Index	Field in vl velocity acceleration deceleration record	Data type
0082 <sub>h</sub>	0	number of entries	UNSIGNED8
	1	Delta speed	UNSIGNED32
	2	Delta time	INTEGER16

## 18.3 Object dictionary entries

### 18.3.1 Objects defined in this chapter

Index	Object	Name	Type	Attr.	M/O
6042 <sub>h</sub>	VAR	vl target velocity	INTEGER16	rw	M
6043 <sub>h</sub>	VAR	vl velocity demand	INTEGER16	ro	M
6044 <sub>h</sub>	VAR	vl control effort	INTEGER16	ro	M
6045 <sub>h</sub>	VAR	vl manipulated velocity	INTEGER16	ro	O
6046 <sub>h</sub>	ARRAY	vl velocity min max amount	UNSIGNED32	rw	M
6047 <sub>h</sub>	ARRAY	vl velocity min max	UNSIGNED32	rw	O
6048 <sub>h</sub>	RECORD	vl velocity acceleration	vl velocity acceleration deceleration record	rw	M
6049 <sub>h</sub>	RECORD	vl velocity deceleration	vl velocity acceleration deceleration record	rw	M
604A <sub>h</sub>	RECORD	vl velocity quick stop	vl velocity acceleration deceleration record	rw	O
604B <sub>h</sub>	ARRAY	vl set-point factor	INTEGER16	rw	O
604C <sub>h</sub>	ARRAY	vl dimension factor	INTEGER32	rw	O
604D <sub>h</sub>	VAR	vl pole number	UNSIGNED8	rw	O
604E <sub>h</sub>	VAR	vl velocity reference	UNSIGNED32	rw	O
604F <sub>h</sub>	VAR	vl ramp function time	UNSIGNED32	rw	O
6050 <sub>h</sub>	VAR	vl slow down time	UNSIGNED32	rw	O
6051 <sub>h</sub>	VAR	vl quick stop time	UNSIGNED32	rw	O
6052 <sub>h</sub>	VAR	vl nominal percentage	INTEGER16	rw	O
6053 <sub>h</sub>	VAR	vl percentage demand	INTEGER16	ro	O
6054 <sub>h</sub>	VAR	vl actual percentage	INTEGER16	ro	O
6055 <sub>h</sub>	VAR	vl manipulated percentage	INTEGER16	ro	O
6056 <sub>h</sub>	ARRAY	vl velocity motor min max amount	UNSIGNED32	rw	O
6057 <sub>h</sub>	ARRAY	vl velocity motor min max	UNSIGNED32	rw	O
6058 <sub>h</sub>	ARRAY	vl frequency motor min max amount	UNSIGNED32	rw	O
6059 <sub>h</sub>	ARRAY	vl frequency motor min max	UNSIGNED32	rw	O

**18.3.2 Objects defined in other chapters**

Index	Object	Name	Type	Chapter
603F <sub>h</sub>	VAR	Error code	UNSIGNED16	ce
6040 <sub>h</sub>	VAR	Controlword	UNSIGNED16	dc
6041 <sub>h</sub>	VAR	Statusword	UNSIGNED16	dc
605A <sub>h</sub>	VAR	Quick stop option code	INTEGER16	dc
605B <sub>h</sub>	VAR	Shut down option code	INTEGER16	dc
605C <sub>h</sub>	VAR	Disable operation option code	INTEGER16	dc
605D <sub>h</sub>	VAR	Halt option code	INTEGER16	dc
6060 <sub>h</sub>	VAR	Modes of operation	INTEGER8	dc
6061 <sub>h</sub>	VAR	Modes of operation display	INTEGER8	dc

**18.4 Object description****18.4.1 Object 6042<sub>h</sub>: *vl target velocity***

The *vl target velocity* is the required velocity of the system. It is multiplied by the *vl dimension factor* and the *vl set-point factor*, if these are implemented. The *vl target velocity* is converted to the unit [rpm] by multiplying the *vl target velocity* by the *vl dimension factor*. The unit of the *vl target velocity* is interpreted as rpm, if the *vl dimension factor* is not implemented or has the value 1.

**OBJECT DESCRIPTION**

INDEX	6042 <sub>h</sub>
Name	vl target velocity
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if vl supported

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	0

**18.4.2 Object 6043<sub>h</sub>: vl velocity demand**

The *vl velocity demand* is the instantaneous velocity provided by the ramp function, scaled to the unit of the *vl target velocity*. The value ranges from -32768 to 32767 (INTEGER16). The parameter could only be read, because it is changed only by the drive.

**OBJECT DESCRIPTION**

INDEX	6043 <sub>h</sub>
Name	vl velocity demand
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if vl supported

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.3 Object 6053<sub>h</sub>: vl percentage demand**

The *vl percentage demand* is calculated on the basis of the *vl velocity demand* by using the percentage function. It is the velocity provided by the ramp function in percent. Accordingly, the *vl percentage demand* is within the same value range as the *vl nominal percentage*. The value ranges from -32768 to 32767 (INTEGER16). The value 16383 corresponds to 100% of the *vl velocity reference*. Accordingly, an indication range of +/- 200% is possible. The parameter is read-only.

**OBJECT DICTIONARY**

INDEX	6053 <sub>h</sub>
Name	vl percentage demand
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.4 Object 6054<sub>h</sub>: vl actual percentage**

The *vl actual percentage* is calculated on the basis of the *vl control effort* by using the percentage function. In this way, the *vl actual percentage* has the same value range as the *vl nominal percentage*. The value ranges from -32768 to 32767 (INTEGER16). The value 16383 corresponds to 100% of the *vl velocity reference*. Therefore, an indication range of +/- 200% is possible.

**OBJECT DESCRIPTION**

INDEX	6054 <sub>h</sub>
Name	vl actual percentage
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.5 Object 6055<sub>h</sub>: vl manipulated percentage**

The *vl manipulated percentage* is calculated on the basis of the *vl manipulated velocity*. In this way, the *vl manipulated percentage* is shown in the same value range as the *vl nominal percentage*. The value ranges from -32768 to 32767 (INTEGER16). The value 16383 corresponds to 100% of the *vl velocity reference*. Therefore, an indication range of +/- 200% is possible.

**OBJECT DESCRIPTION**

INDEX	6055 <sub>h</sub>
Name	vl manipulated percentage
Object Code	VAR
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.6 Object 604E<sub>h</sub>: vl velocity reference**

This parameter serves to represent *velocity values* (set-points, actual values and ramps) as relative values. If the *vl velocity reference* is modified, the ramps slopes, if objects *vl ramp function time*, *vl slow down time* or *vl quick stop time* are implemented, are changed relative to the change in the *vl velocity reference*.

This parameter has the same unit as the *vl target velocity* and the following value range: 0 .. 4.294.967.295 (UNSIGNED32).

## OBJECT DESCRIPTION

INDEX	604E <sub>h</sub>
Name	vl velocity reference
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

## CONVERTING PERCENTAGES TO VELOCITY VALUES

$$\text{Velocity value} = \frac{\text{Percentage} \cdot \text{vl velocity reference}}{3FFF_h}$$

## CONVERTING VELOCITY VALUES TO PERCENTAGES

$$\text{Percentage} = \frac{\text{Velocity value} \cdot 3FFF_h}{\text{vl velocity reference}}$$

**18.4.7 Object 604C<sub>h</sub>: vl dimension factor**

The *vl dimension factor* is generated by division using the sub-parameter numerator and the sub-parameter denominator. These parameter have a value ranging from –2.147.483.648 to 2.147.483.647 (INTEGER32), but except the value 0!

The *vl dimension factor* serves to include gearing in calculation or serves to scale the frequencies or specific units of the user. It influences the specified set-point, the velocity limit and the ramp function as well as the output variables of the speed function.

## OBJECT DESCRIPTION

INDEX	604C <sub>h</sub>
Name	vl dimension factor
Object Code	ARRAY
Data Type	INTEGER32
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	vl dimension factor numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	1

Sub-Index	2
Description	vl dimension factor denominator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER32
Default Value	1

#### CALCULATING THE VL DIMENSION FACTOR

Every user specific speed consists of a specific unit referred to a specific unit of time (e.g. 1/sec, bottles/min, m/sec,...).

The purpose of the dimension factor is to convert this specific unit to the revolutions/minute unit.

$$\text{Specific unit} \cdot \text{DF} = 1 \text{ revolution (motor shaft)}$$

$$I \cdot \text{DF} = O$$

I = vl target velocity expressed as the user's specific speed

Input value of the factor function

Unit of I: [I] = Specific unit

O = Speed value in [rpm]

Output value of the factor function

$$\text{Unit of O: } [O] = \frac{1}{\text{min}} = \frac{\text{Revolution}}{\text{min}}$$

DF = Dimension factor

$$\text{Unit of DF: } [\text{DF}] = \frac{1}{\text{Specific unit}} * \frac{1}{\text{min}}$$

**18.4.8 Object 604B<sub>n</sub>: vl set-point factor**

The *vl set-point factor* is generated by division, using a numerator sub-parameter and a denominator sub-parameter. These sub-parameters have no unit and have values within a range from -32768 to 32767 (INTEGER16), but excluding the value 0!

The *vl set-point factor* serves to modify the resolution or directing range of the specified set-point. It is included in calculation of the specified set-point and the output variables of the speed function only.

**OBJECT DESCRIPTION**

INDEX	604B <sub>n</sub>
Name	vl set-point factor
Object Code	ARRAY
Data Type	INTEGER16
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	vl set-point factor numerator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	1

Sub-Index	2
Description	vl set-point factor denominator
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	1



**18.4.9 Object 604D<sub>h</sub>: *vl pole number***

The user must describe the *vl pole number* parameter with a value corresponding to the number of poles belonging to the connected motor. This parameter has no unit. The value range depends on the manufacturer-specific need and is represented as UNSIGNED8. If the Object *vl pole number* does fit for the desired type of motor, this object could left out or set to value 2.

**OBJECT DESCRIPTION**

INDEX	604D <sub>h</sub>
Name	<i>vl pole number</i>
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	possible
Value Range	Unsigned8
Default Value	No

If the number of pole pairs is known, the number of poles is:  $vl\ pole\ number = 2 \cdot pole\ pairs$

**CONVERTING VELOCITY VALUES TO FREQUENCY VALUES**

$$Frequency = \frac{Velocity \cdot vl\ pole\ number}{60 \cdot 2}$$

**CONVERTING FREQUENCY VALUES TO VELOCITY VALUES**

$$Velocity = \frac{Frequency \cdot 60 \cdot 2}{vl\ pole\ number}$$

**18.4.10 Object 6046<sub>h</sub>: *vl velocity min max amount***

The *vl velocity min max amount* parameter is composed of the *vl velocity min amount* and *vl velocity max amount* sub-parameter. These sub-parameters don't have units and have values within a range from 0 to 4.294.967.295 (UNSIGNED32):

The *vl velocity max amount* sub-parameter is mapped internally to the *vl velocity max pos* and *vl velocity max neg* values. The *vl velocity min amount* sub-parameter is mapped internally to the *vl velocity min pos* and *vl velocity min neg* values.

Only the positive values are returned if the *vl velocity min max amount* parameter is read out.

**OBJECT DESCRIPTION**

INDEX	6046 <sub>h</sub>
Name	<i>vl velocity min max amount</i>
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Mandatory, if vl supported

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	vl velocity min amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	vl velocity max amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl velocity min max amount*

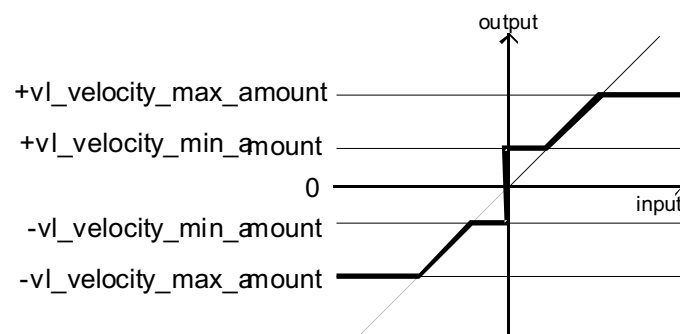


Figure 46: *vl velocity min max amount* transfer characteristic

**18.4.11 Object 6047<sub>h</sub>: vl velocity min max**

The *vl velocity min max* parameter is composed of the *vl velocity min pos*, *vl velocity max pos*, *vl velocity min neg* and *vl velocity max neg* sub-parameter. These sub-parameter have no units and have values within a range from 0 to 4.294.967.295 (UNSIGNED32).

The sub-parameter are mapped internally to the corresponding values.

**OBJECT DESCRIPTION**

INDEX	6047 <sub>h</sub>
Name	vl velocity min max
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1
Description	vl velocity min pos
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	vl velocity max pos
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	3
Description	vl velocity min neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	4
Description	vl velocity max neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl velocity min max*

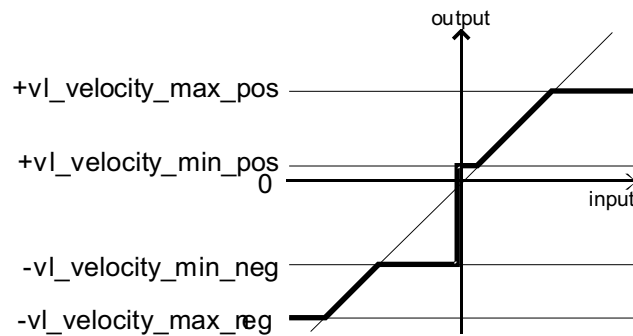


Figure 47: *vl velocity min max* transfer characteristic

#### 18.4.12 Object 6058<sub>h</sub>: *vl frequency motor min max amount*

The frequency parameter of the *vl frequency motor min max amount* objects are mapped internally to the parameter of the corresponding speed objects.

##### OBJECT DESCRIPTION

INDEX	6058 <sub>h</sub>
Name	<i>vl frequency motor min max amount</i>
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	vl frequency motor min amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	vl frequency motor max amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl frequency motor min max amount*

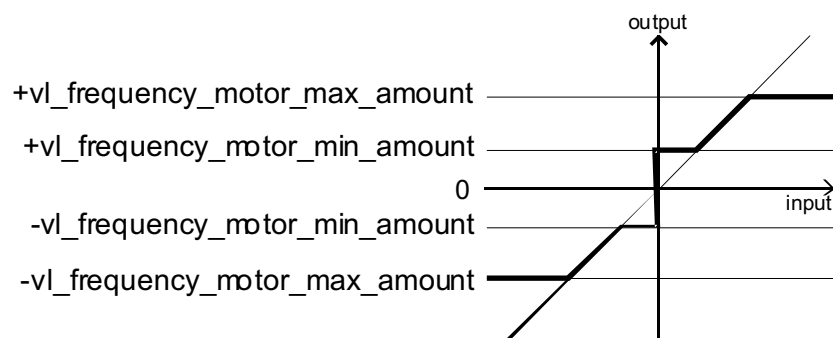


Figure 48: *vl frequency motor min max amount* transfer characteristic

**18.4.13 Object 6059<sub>h</sub>: *vl frequency motor min max***

The frequency parameter of the *vl frequency motor min max* objects are mapped internally to the parameter of the corresponding speed objects.

**OBJECT DESCRIPTION**

INDEX	6059 <sub>h</sub>
Name	<i>vl frequency motor min max</i>
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1
Description	<i>vl frequency motor min pos</i>
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	<i>vl frequency motor max pos</i>
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	3
Description	vl frequency motor min neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	4
Description	vl frequency motor max neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl velocity min max*

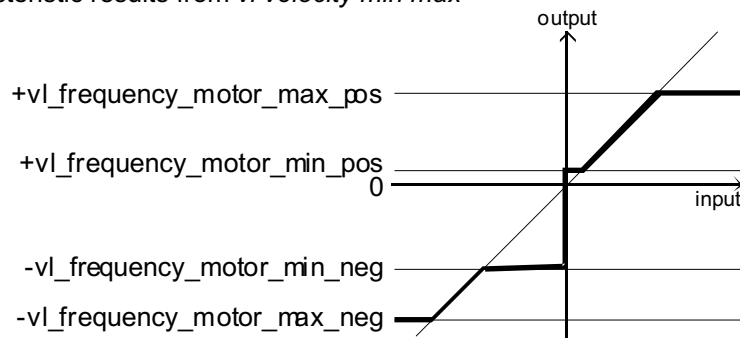


Figure 49: *vl velocity min max* transfer characteristic

#### 18.4.14 Object 6056<sub>h</sub>: *vl velocity motor min max amount*

The *vl velocity motor min max amount* parameter is composed of the *vl velocity motor min amount* and *vl velocity motor max amount* parameter. These sub-parameters have the unit [(1/1000) rpm] and values within a range from 0 to 4.294.967.295 [(1/1000) rpm] (UNSIGNED32). This results in a limiting range from 0 to 4.294.967.295 [rpm].

The *vl velocity motor max amount* sub-parameter is mapped internally to the *vl velocity motor max pos* and *vl velocity motor max neg* values. The *vl velocity motor min amount* sub-parameter is mapped internally to the *vl velocity motor min pos* and *vl velocity motor min neg* values.

Only the positive values are returned, if the *vl velocity motor min max amount* parameter is read.

##### OBJECT DESCRIPTION

INDEX	6056 <sub>h</sub>
Name	vl velocity motor min max amount
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	vl velocity motor min amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	vl velocity motor max amount
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl velocity motor min max amount*

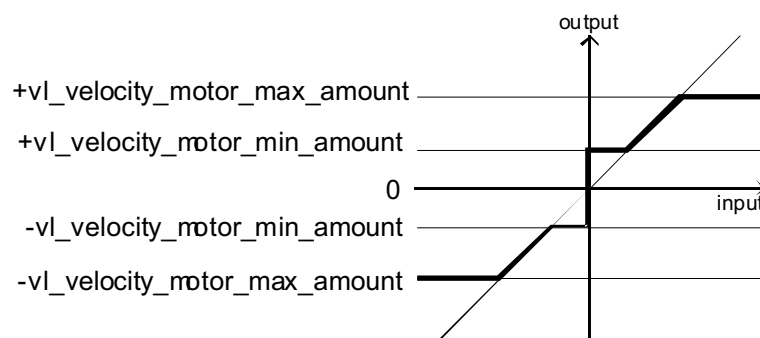


Figure 50: *vl velocity motor min max amount* transfer characteristic



**18.4.15 Object 6057<sub>h</sub>: vl velocity motor min max**

The *vl velocity motor min max* parameter is composed of the *vl velocity motor min pos*, *vl velocity motor max pos*, *vl velocity motor min neg* and *vl velocity motor max neg* sub-parameter. These sub-parameters have the unit [1/(1000 min)] and values within a range from 0 ... 4.294.967.295 [1/(1000 min)] (UNSIGNED32). This results in a limiting range from 0 ... 4.294.967 [rpm].

The sub-parameter are mapped internally to the corresponding values.

**OBJECT DESCRIPTION**

INDEX	6057 <sub>h</sub>
Name	vl velocity motor min max
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	4
Default Value	4

Sub-Index	1
Description	vl velocity motor min pos
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	vl velocity motor max pos
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	3
Description	vl velocity motor min neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	4
Description	vl velocity motor max neg
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

This transfer characteristic results from *vl velocity min max*

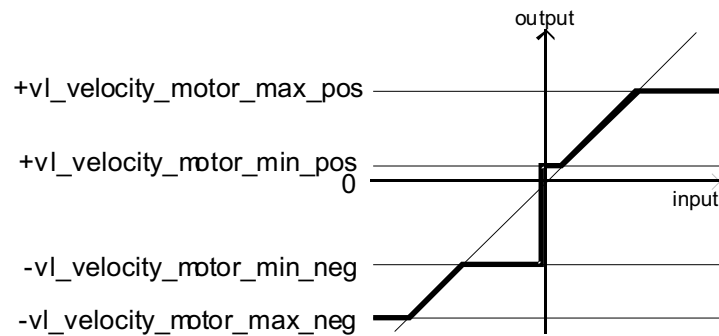


Figure 51: *vl velocity min max* transfer characteristic

#### LIMIT-VALUE

The Limit-value message is generated, if the input value of the *velocity motor limit function* results in a value outside of the operating range of the *velocity motor limit function*. The Limit-value message is mapped as one bit in the *statusword*.

#### 18.4.16 Object 6048<sub>h</sub>: vl velocity acceleration

The *vl velocity acceleration* parameter specifies the slope of the acceleration ramp. It is generated as the quotient of the *delta speed* and *delta time* sub-parameter.

#### OBJECT DESCRIPTION

INDEX	6048 <sub>h</sub>
Name	vl velocity acceleration
Object Code	RECORD
Data Type	vl velocity acceleration deceleration record (0082 <sub>h</sub> )
Category	Conditional; Mandatory, if vl supported

## ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	Delta time
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

## DELTA SPEED

The *delta speed* has the same unit as the *vl target velocity*. This sub-parameter has the value range: 0 ... 4.294.967.295 (UNSIGNED32).

## DELTA TIME

This sub-parameter is specified in sec and has the value range: 0 ... 65.535 [sec] (UNSIGNED16).

This function directly follows the set-point if the parameter 0 is defined for the *delta time* value.

$$vl \text{ velocity acceleration} = \frac{\text{delta speed}}{\text{delta time}} = a_B$$

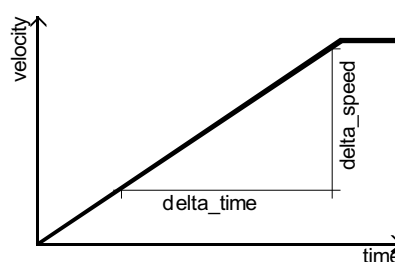


Figure 52: *vl velocity acceleration* transfer characteristic

**18.4.17 Object 6049<sub>h</sub>: vl velocity deceleration**

The *vl velocity deceleration* parameter specifies the slope of the deceleration ramp. It is generated as the quotient of the *delta speed* and *delta time* sub-parameter.

**OBJECT DESCRIPTION**

INDEX	6049 <sub>h</sub>
Name	vl velocity deceleration
Object Code	RECORD
Data Type	vl velocity acceleration deceleration record (0082 <sub>h</sub> )
Category	Conditional; Mandatory, if vl supported

**ENTRY DESCRIPTION**

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	Delta time
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**DELTA SPEED**

The *delta speed* has the same unit as the *vl target velocity*. This sub-parameter has the value range: 0 ... 4.294.967.295 (UNSIGNED32).

**DELTA TIME**

This sub-parameter is specified in sec and has the value range: 0 ... 65.535 [sec] (UNSIGNED16).

This function directly follows the set-point, if the value 0 is defined for the *delta time* parameter.

$$vl \text{ velocity deceleration} = \frac{\text{delta speed}}{\text{delta time}} = a_v$$

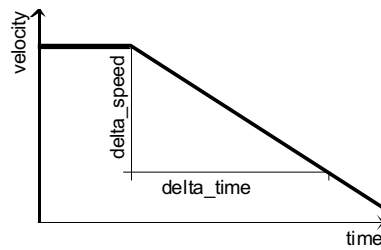


Figure 53: *vl velocity deceleration* transfer characteristic

#### 18.4.18 Object 604A<sub>n</sub>: *vl velocity quick stop*

The *vl velocity quick stop* parameter specifies the slope of the quick stop ramp. It is generated as the quotient of the *delta speed* and *delta time* sub-parameter.

##### OBJECT DESCRIPTION

INDEX	604A <sub>n</sub>
Name	vl velocity quick stop
Object Code	RECORD
Data Type	vl velocity acceleration deceleration record (0082 <sub>n</sub> )
Category	Optional

##### ENTRY DESCRIPTION

Sub-Index	0
Description	number of entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2
Default Value	2

Sub-Index	1
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED16
Default Value	No

**DELTA SPEED**

The *delta speed* has the same unit as the *vl target velocity*. This sub-parameter has the value range: 0 ... 4.294.967.295 (UNSIGNED32).

**DELTA\_TIME**

This sub-parameter is specified in sec and has the value range: 0 ... 65.535 [sec] (UNSIGNED16).

This function directly follows the set-point, if the parameter 0 is defined for the *delta time* value.

$$velocity\ quick\ stop = \frac{delta\ speed}{delta\ time} = a_s$$

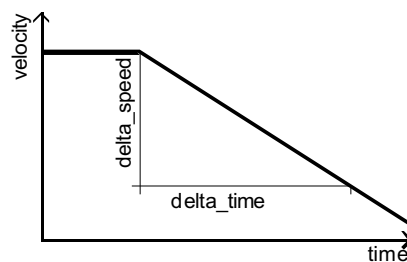


Figure 54: *vl velocity quick stop* transfer characteristic

**18.4.19 Object 604F<sub>h</sub>: vl ramp function time**

The *vl ramp function time* specifies the time during which the drive starts up from zero to the *vl velocity reference*.

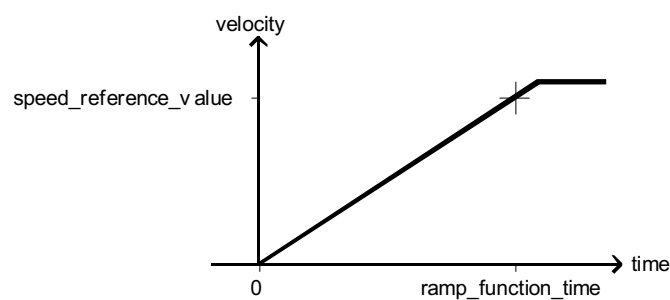


Figure 55: *vl ramp function time* transfer characteristic

This parameter is specified in ms and has the value range: 0 ... 4.294.967.295 [ms] (UNSIGNED32).

By setting the parameter 0 for the *vl ramp function time*, the ramp becomes infinite and the reference variable directly follows the set-point.

## OBJECT DESCRIPTION

INDEX	604F <sub>h</sub>
Name	vl ramp function time
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**18.4.20 Object 6050<sub>h</sub>: vl slow down time**

The *vl slow down time* specifies the time during which the drive slows down from *vl velocity reference* to zero.

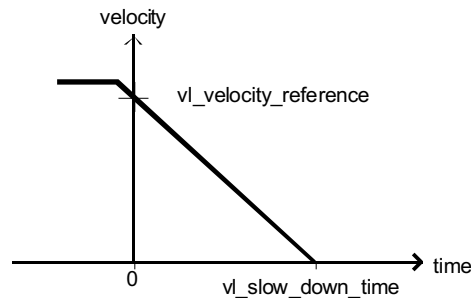


Figure 56: *vl slow down time* transfer characteristic

This parameter is specified in ms and has the value range: 0 ... 4.294.967.295 [ms] (UNSIGNED32).

By defining the parameter 0 for the *vl slow down time*, the ramp becomes infinite and the reference variable directly follows the set-point.

## OBJECT DESCRIPTION

INDEX	6050 <sub>h</sub>
Name	vl slow down time
Object Code	VAR
Data Type	UNSIGNED32

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**18.4.21 Object 6051<sub>h</sub>: vl quick stop time**

The *vl quick stop time* specifies the time during which the drive slows down from *vl velocity reference* to zero in the QUICK STOP ACTIVE state.

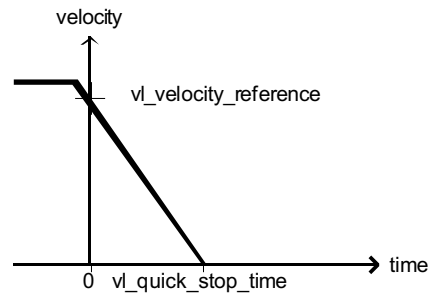


Figure 57: *vl quick stop time* transfer characteristic

This parameter is specified in ms and has the value range: 0 ... 4.294.967.295 [ms] (UNSIGNED32).

By defining the parameter 0 for the *vl quick stop time*, the ramp becomes infinite and the reference variable directly follows the set-point.

**OBJECT DESCRIPTION**

INDEX	6051 <sub>h</sub>
Name	vl quick stop time
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

**ENTRY DESCRIPTION**

Access	rw
PDO Mapping	Possible
Value Range	UNSIGNED32
Default Value	No

**18.4.22 Object 6044<sub>h</sub>: vl control effort**

The *vl control effort* is the velocity at the motor spindle or load, scaled to the unit of the *vl target velocity*. Depending on the system, velocity deviations may occur between the *vl control effort* and the physical velocity. For simple drives without closed loop control or observer this value reads the object *vl velocity demand*. The value ranges from –32.768 to 32.767 (INTEGER16).

**OBJECT DESCRIPTION**

INDEX	6044 <sub>h</sub>
Name	vl control effort
Object Code	VAR
Data Type	INTEGER16
Category	Conditional; Mandatory, if vl supported



## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.23 Object 6045<sub>h</sub>: *vl manipulated velocity***

The *vl manipulated velocity* is the velocity of the motor spindle or load with a compensation value, scaled to the unit of the *vl target velocity*. The compensation value is generated by the controller/control function. The value ranges from –32.768 to 32.767 (INTEGER16).

## OBJECT DESCRIPTION

INDEX	6045 <sub>h</sub>
Name	vl manipulated velocity
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	ro
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

**18.4.24 Object 6052<sub>h</sub>: *vl nominal percentage***

The *vl nominal percentage* is converted by the percent function to a velocity value. The *vl nominal percentage* has no unit (better (100 / 16.383) %). Its value ranges from –32.768 to 32.767 (INTEGER16). The value 16.383 corresponds to 100% of the *vl velocity reference*. Accordingly, a total range of the manipulated variable amounting to +/- 200% is possible.

## OBJECT DESCRIPTION

INDEX	6052 <sub>h</sub>
Name	vl nominal percentage
Object Code	VAR
Data Type	INTEGER16
Category	Optional

## ENTRY DESCRIPTION

Access	rw
PDO Mapping	Possible
Value Range	INTEGER16
Default Value	No

## 18.5 Functional description

### 18.5.1 Percentage function

The percentage function serves to convert percentages to velocity values (see Figure 58) and vice versa (see Figure 59).

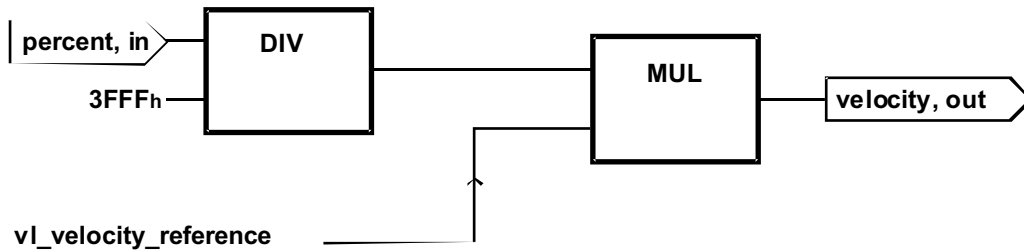


Figure 58: Percentage function

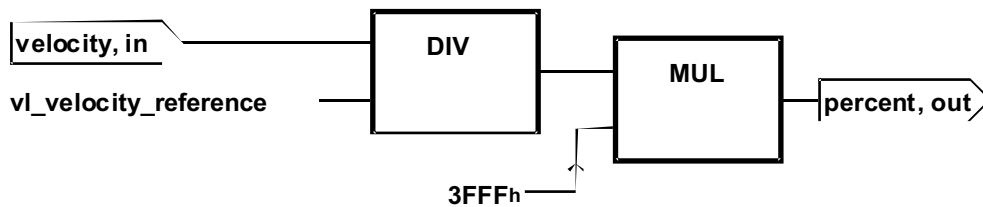


Figure 59: Reverse percentage function

### 18.5.2 Factor function and reverse factor function

The factor function multiplies the input variables by the assigned factors.

- The *vl target velocity* is multiplied by the *vl dimension factor* and the *vl set-point factor*.
- The values of the velocity limit and the values for the ramp function are only multiplied by the *vl dimension factor*.

The factor has a value of 1, if it is not implemented.

The factor function for two factors is built of two function in series connection.

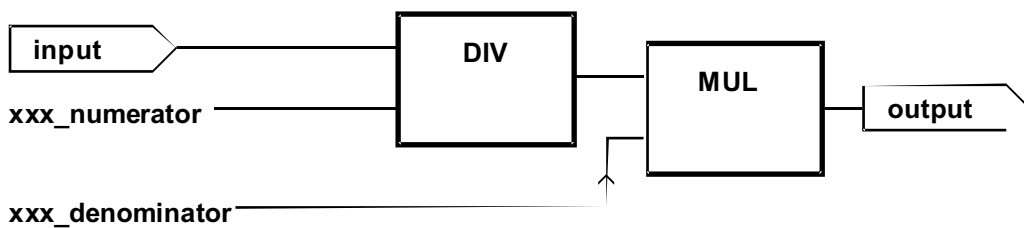


Figure 60: Factor function

The reverse factor function divides the input variables by the assigned factors.

- The output variables of the velocity mode are calculated by division with the *vl dimension factor* and the *vl set-point factor* and therefore returned to the scaling of the specified set-point.

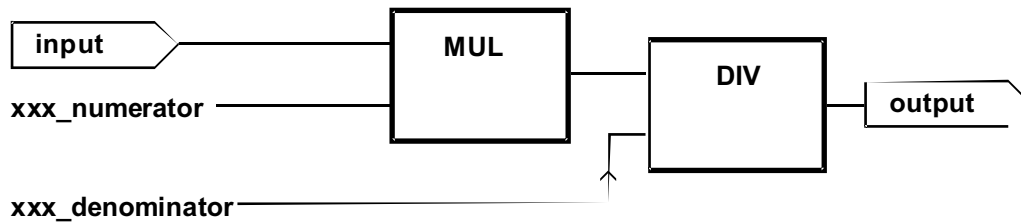


Figure 61: Reverse factor function

### 18.5.3 Pole number function

The pole number function serves to convert frequency values to velocity values

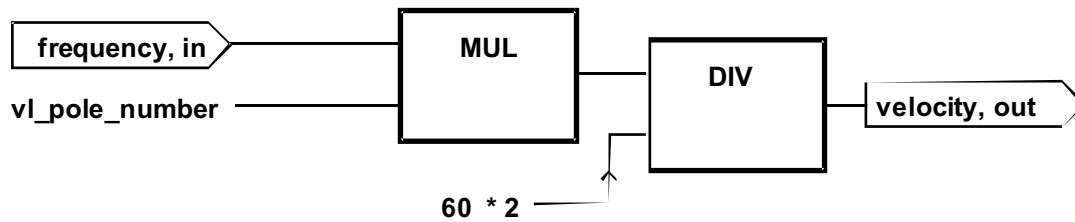


Figure 62: Pole number function

and vice versa

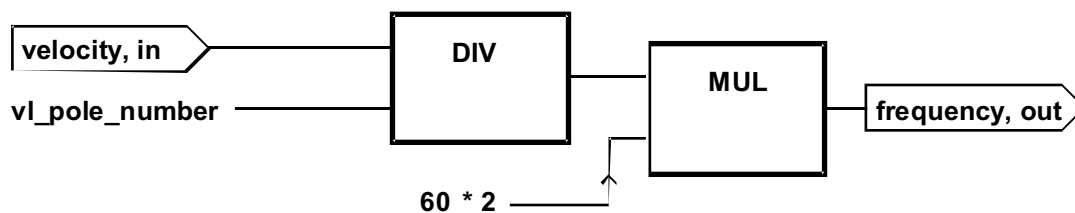


Figure 63: Reverse pole number function

### 18.5.4 Velocity limit function

The velocity limit defines the valid velocity range for the drive. Limits could be specified in the user specific units by including the *vl dimension factor* in the speed limit.

#### LIMIT-VALUE

The Limit-value message is generated if the input value of the speed limit results in a value outside the speed limit's operating range. The Limit-value message is mapped as one bit in the *statusword*.

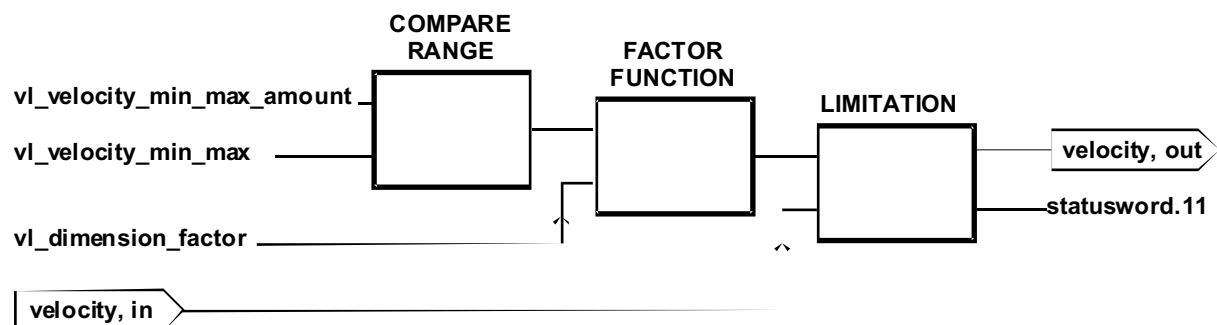


Figure 64: Velocity limit function

### 18.5.5 Velocity motor limit function

The velocity motor limit function limits the motor velocity range. This parameter has a safety function that ensures that the range of the set value of motor velocity cannot be exceeded inadvertently by a modification of a factor.

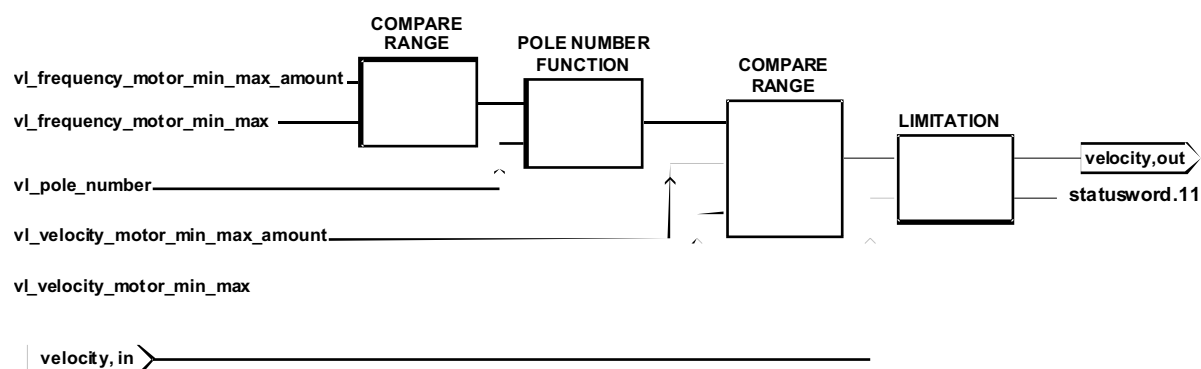


Figure 65: Velocity motor limit function

### 18.5.6 Ramp function

The ramp function is used to limit the increase and decrease of velocity. The velocity output is equal to the input as long as the changes are below  $a_{B_{min}}$ ,  $a_{V_{min}}$  or  $a_{S_{min}}$ .

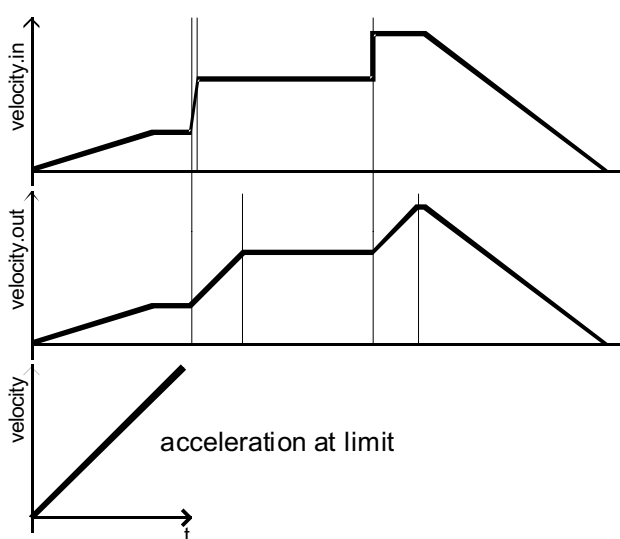


Figure 66: Velocity profile

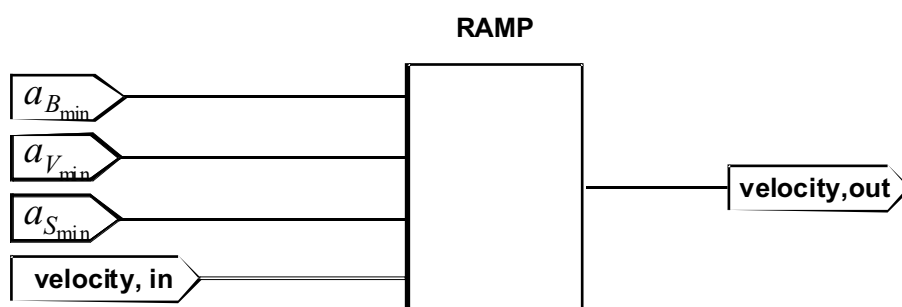


Figure 67: Ramp function

The internal ramp values  $a_{B_{\min}}$  and  $a_{V_{\min}}$  directly consist of the *vl velocity acceleration* and *vl velocity deceleration* parameter.

The internal ramp values  $a_{B_{\min}}$ ,  $a_{V_{\min}}$  and  $a_{S_{\min}}$  are the output values of the ramp min function, weighted with the *vl dimension factor*.

### 18.5.7 Ramp min function

The ramp min function selects the minimal change of velocity.

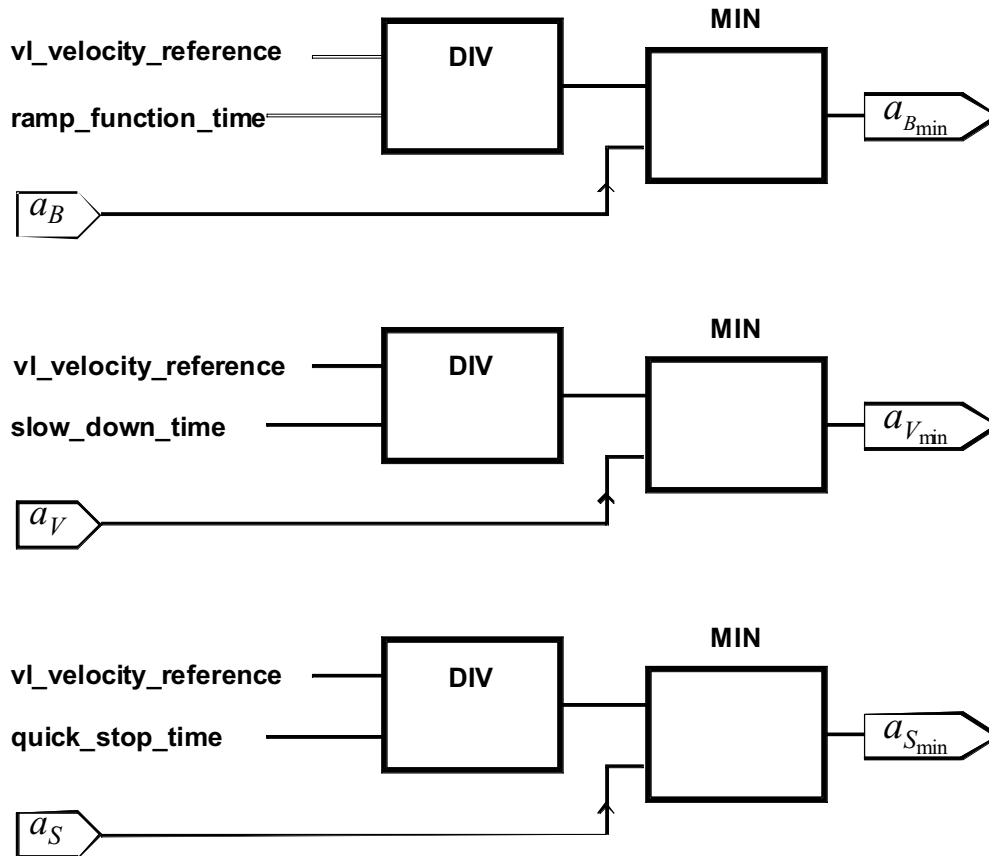


Figure 68: Ramp min function

The internal ramp values ( $a_{B_{\min}}$ ,  $a_{V_{\min}}$ ,  $a_{S_{\min}}$ ) are recalculated as follows, if one of the input parameter for the ramp function is modified:

$$a_{B_{\min}} = \text{MIN} \left( a_B, \frac{\text{vl velocity reference}}{\text{vl ramp function time}} \right)$$

$$a_{V_{\min}} = \text{MIN} \left( a_V, \frac{\text{vl velocity reference}}{\text{vl slow down time}} \right)$$

$$a_{S_{\min}} = \text{MIN} \left( a_S, \frac{\text{vl velocity reference}}{\text{vl quick stop time}} \right)$$

The ramp min function selects the lower respective value of the slopes.

### 18.5.8 Reference calculation

This sub-function decides on the set-point processing. The set-point value may be given as an percentage and (or) as an absolute value. Therefore two objects are defined in this profile. The first object is the *vl target velocity* containing the absolute set-point value. This is an mandatory object for every drive. Some drives may have implemented the object *vl nominal percentage*, which consist of an percentage set-point value. So these manufactures have to specify how this two values are handled

within the drive. Most profile implementations will add the percentage and the absolute set-point value to calculate the internal set-point. It is also possible to use only one value. Then the last written object is used internally.

#### **18.5.9 Closed open loop control function**

On the basis of the *vl control effort*, the controller/control function returns the *vl control effort* and the *vl manipulated velocity*.

Depending on realization of the function, the *vl control effort* is the *vl control effort* or a calculated or measured *vl control effort*.

Depending on realization of the function, the *vl manipulated velocity* is the *vl control effort* or a calculated *speed output*.

# 19APPENDIX

## 19.1 Object dictionary by chapter

### 19.1.1 Common entries

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional							Units	Description
							all	hm	pp	pc	ip	pV	tq	VI	
6007 <sub>h</sub>	-	VAR	Abort connection option code	INTEGER16	rw	Possible	O								-
603F <sub>h</sub>	-	VAR	Error code	UNSIGNED16	ro	Possible	O								-
6402 <sub>h</sub>	-	VAR	Motor type	UNSIGNED16	rw	Possible	O								-
6403 <sub>h</sub>	-	VAR	Motor catalog number	VISIBLE_STRING	rw	No	O								-
6404 <sub>h</sub>	-	VAR	Motor manufacturer	VISIBLE_STRING	rw	No	O								-
6405 <sub>h</sub>	-	VAR	http motor catalog address	VISIBLE_STRING	rw	No	O								-
6406 <sub>h</sub>	-	VAR	Motor calibration date	TIME_OF_DAY	rw	Possible	O								-
6407 <sub>h</sub>	-	VAR	Motor service period	UNSIGNED32	rw	Possible	O								-
6410 <sub>h</sub>	-	RECORD	Motor data	(manufacturer specific)	rw	-	O								-
6502 <sub>h</sub>	-	VAR	Supported drive modes	UNSIGNED32	ro	Possible	O								-
6503 <sub>h</sub>	-	VAR	Drive catalog number	VISIBLE_STRING	rw	No	O								-
6504 <sub>h</sub>	-	VAR	Drive manufacturer	VISIBLE_STRING	rw	No	O								-
6505 <sub>h</sub>	-	VAR	http drive catalog address	VISIBLE_STRING	rw	No	O								-
6510 <sub>h</sub>	-	RECORD	Drive data	(manufacturer specific)	rw	-	O								-
60FD <sub>h</sub>	-	VAR	Digital inputs	UNSIGNED32	ro	Possible	O								-
60FE <sub>h</sub>	-	ARRAY	Digital outputs	UNSIGNED32	-	-	O								-
	1		Physical outputs		rw	Possible	M								-
	2		Bitmask		rw	Possible	O								-

## 19.1.2 Device control

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	VI		
6040 <sub>h</sub>	-	VAR	Controlword	UNSIGNED16	rw	Possible	M								-	
6041 <sub>h</sub>	-	VAR	Statusword	UNSIGNED16	ro	Possible	M								-	
605B <sub>h</sub>	-	VAR	Shutdown option code	INTEGER16	rw	No	O								-	
605C <sub>h</sub>	-	VAR	Disable operation option code	INTEGER16	rw	No	O								-	
605A <sub>h</sub>	-	VAR	Quick stop option code	INTEGER16	rw	No	O								-	
605D <sub>h</sub>	-	VAR	Halt option code	INTEGER16	rw	No	O								-	
605E <sub>h</sub>	-	VAR	Fault reaction option code	INTEGER16	rw	No	O								-	
6060 <sub>h</sub>	-	VAR	Modes of operation	INTEGER8	rw	Possible	M								-	
6061 <sub>h</sub>	-	VAR	Modes of operation display	INTEGER8	ro	Possible	M								-	

## 19.1.3 Factor group

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	VI		
6089 <sub>h</sub>	-	VAR	Position notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608A <sub>h</sub>	-	VAR	Position dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608B <sub>h</sub>	-	VAR	Velocity notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608C <sub>h</sub>	-	VAR	Velocity dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608D <sub>h</sub>	-	VAR	Acceleration notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608E <sub>h</sub>	-	VAR	Acceleration dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608F <sub>h</sub>	-	ARRAY	Position encoder resolution	UNSIGNED32	-	-			O		O	O	O		inc / rev	
	1		Encoder increments		rw	Possible			M		M	M	M		inc	
	2		Motor revolutions		rw	Possible			M		M	M	M		rev	
6090 <sub>h</sub>	-	ARRAY	Velocity encoder resolution	UNSIGNED32	-	-			O		O	O	O		inc / rev	
	1		Encoder increments per second		rw	Possible			M		M	M	M		inc / s	
	2		Motor revolutions per second		rw	Possible			M		M	M	M		rev / s	



Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vI		
6091 <sub>h</sub>	-	ARRAY	Gear ratio	UNSIGNED32	-	-			O		O	O	O		-	
	1		Motor revolutions		rw	Possible			M		M	M	M		rev	
	2		Shaft revolutions		rw	Possible			M		M	M	M		rev	
6092 <sub>h</sub>	-	ARRAY	Feed constant	UNSIGNED32	-	-			O		O	O	O		pos units / rev	
	1		Feed		rw	Possible			M		M	M	M		pos units	
	2		Shaft revolutions		rw	Possible			M		M	M	M		rev	
6093 <sub>h</sub>	-	ARRAY	Position factor	UNSIGNED32	-	-			O	O	O				-	
	1		Numerator		rw	Possible			M	M	M				-	
	2		Feed constant		rw	Possible			M	M	M				-	
6094 <sub>h</sub>	-	ARRAY	Velocity encoder factor	UNSIGNED32	-	-			O	O		O			-	
	1		Numerator		rw	Possible			M	M		M			-	
	2		Divisor		rw	Possible			M	M		M			-	
6095 <sub>h</sub>	-	ARRAY	Velocity factor 1	UNSIGNED32	-	-			O	O	O	O	O		-	
	1		Numerator		rw	Possible			M	M	M	M	M		-	
	2		Divisor		rw	Possible			M	M	M	M	M		-	
6096 <sub>h</sub>	-	ARRAY	Velocity factor 2	UNSIGNED32	-	-			O		O	O	O		-	
	1		Numerator		rw	Possible			M		M	M	M		-	
	2		Divisor		rw	Possible			M		M	M	M		-	
6097 <sub>h</sub>	-	ARRAY	Acceleration factor	UNSIGNED32	-	-			O	O	O	O	O		-	
	1		Numerator		rw	Possible			M	M	M	M	M		-	
	2		Divisor		rw	Possible			M	M	M	M	M		-	
607E <sub>h</sub>	-	VAR	Polarity	UNSIGNED8	rw	Possible			O	O	O	O	O		-	

## 19.1.4 Profile position mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional							Units	Description
							all	hm	pp	pc	ip	pV	tq	vl	
607A <sub>h</sub>	-	VAR	Target position	INTEGER32	rw	Possible			M	M					pos units
607B <sub>h</sub>	-	ARRAY	Position range limit	INTEGER32	-	-			O	O					-
			Min position range limit						M	M					pos units
			Max position range limit						M	M					pos units
607D <sub>h</sub>	-	ARRAY	Software position limit	INTEGER32	-	-			O	O					-
			Min position limit						M	M					pos units
			Max position limit						M	M					pos units
607F <sub>h</sub>	-	VAR	Max profile velocity	UNSIGNED32	rw	Possible			O		O				speed units
6080 <sub>h</sub>	-	VAR	Max motor speed	UNSIGNED32	rw	Possible			O			O			rpm
6081 <sub>h</sub>	-	VAR	Profile velocity	UNSIGNED32	rw	Possible			M			M			speed units
6082 <sub>h</sub>	-	VAR	End velocity	UNSIGNED32	rw	Possible			O						speed units
6083 <sub>h</sub>	-	VAR	Profile acceleration	UNSIGNED32	rw	Possible			M			M			acc units
6084 <sub>h</sub>	-	VAR	Profile deceleration	UNSIGNED32	rw	Possible			M			M			acc units
6085 <sub>h</sub>	-	VAR	Quick stop deceleration	UNSIGNED32	rw	Possible			O			O			acc units
6086 <sub>h</sub>	-	VAR	Motion profile type	INTEGER16	rw	Possible			M			M			-
60C5 <sub>h</sub>	-	VAR	Max acceleration	UNSIGNED32	rw	Possible			O						acc units
60C6 <sub>h</sub>	-	VAR	Max deceleration	UNSIGNED32	rw	Possible			O						acc units

## 19.1.5 Homing mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
607C <sub>h</sub>	-	VAR	Home offset	INTEGER32	rw	Possible		O		O					pos units	
6098 <sub>h</sub>	-	VAR	Homing method	INTEGER8	rw	Possible		M							-	
6099 <sub>h</sub>	-	ARRAY	Homing speeds	UNSIGNED32	-	-		M								
			Speed during search for switch					M							vel units	
			Speed during search for zero					M							vel units	
	2		Speed during search for zero			Possible		M							acc units	
			Speed during search for zero			Possible		M								
609A <sub>h</sub>	-	VAR	Homing acceleration	UNSIGNED32	rw	Possible		O								

## 19.1.6 Position control function

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
6062 <sub>h</sub>	-	VAR	Position demand value	INTEGER32	ro	Possible				O					pos units	
6063 <sub>h</sub>	-	VAR	Position actual value*	INTEGER32	ro	Possible				O		O			inc	
6064 <sub>h</sub>	-	VAR	Position actual value	INTEGER32	ro	Possible		O	O	M	O		O		pos units	
6065 <sub>h</sub>	-	VAR	Following error window	UNSIGNED32	rw	Possible				O					pos units	
6066 <sub>h</sub>	-	VAR	Following error time out	UNSIGNED16	rw	Possible				O					ms	
6067 <sub>h</sub>	-	VAR	Position window	UNSIGNED32	rw	Possible				O					pos units	
6068 <sub>h</sub>	-	VAR	Position window time	UNSIGNED16	rw	Possible				O					ms	
60F4 <sub>h</sub>	-	VAR	Following error actual value	INTEGER32	ro	Possible				O					pos units	
60FA <sub>h</sub>	-	VAR	Control effort	INTEGER32	ro	Possible				O					-	
60FB <sub>h</sub>	-	RECORD	Position control parameter set	(manufacturer specific)	rw	-				O					-	
60FC <sub>h</sub>	-	VAR	Position demand value*	INTEGER32	ro	Possible				O					inc	

## 19.1.7 Interpolated position mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional							Units	Description	
							all	hm	pp	pc	ip	pV	tq			vl
60C0 <sub>h</sub>	-	VAR	Interpolation sub mode select	INTEGER16	rw	Possible					O				-	
60C1 <sub>h</sub>	-	ARRAY	Interpolation data record	(see 60C0 <sub>h</sub> )	-	-					O				-	
	1		X <sub>1</sub>		rw	Possible					M				-	
	2		X <sub>2</sub>		rw	Possible					O				-	
	...		.....		...	.....					...					
	N		X <sub>N</sub>		rw	Possible					O				-	
60C2 <sub>h</sub>	-	RECORD	Interpolation time period	Interpolation time period record	-	-					O				-	
	1		ip time units	UNSIGNED8	rw	Possible					M				10 <sup>6</sup> time index s	
	2		ip time index	INTEGER8	rw	Possible					M				-	
60C3 <sub>h</sub>	-	ARRAY	Interpolation sync definition	UNSIGNED8	-	-					O				-	
	1	Synchronize on group	rw		Possible					M					-	
	2	ip sync every n event	rw		Possible					M					counts	
60C4 <sub>h</sub>	-	RECORD	Interpolation data configuration	Interpolation data configuration record	-	-					O				-	
	1		Maximum buffer size	UNSIGNED32	ro	Possible					M				-	
	2		Actual buffer size	UNSIGNED32	rw	Possible					M				-	
	3		Buffer organization	UNSIGNED8	rw	Possible					M				-	
	4		Buffer position	UNSIGNED16	rw	Possible					M				-	
	5		Size of data record	UNSIGNED8	wo	Possible					M				-	
	6		Buffer clear	UNSIGNED8	wo	Possible					M				-	

## 19.1.8 Profile velocity mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description	
							all	hm	pp	pc	ip	pv	tq	vl			
6069 <sub>h</sub>	-	VAR	Velocity sensor actual value	INTEGER32	ro	Possible							M			inc / s	
606A <sub>h</sub>	-	VAR	Sensor selection code	INTEGER16	rw	Possible							O			-	
606B <sub>h</sub>	-	VAR	Velocity demand value	INTEGER32	ro	Possible							M			vel units	
606C <sub>h</sub>	-	VAR	Velocity actual value	INTEGER32	ro	Possible							M			vel units	
606D <sub>h</sub>	-	VAR	Velocity window	UNSIGNED16	rw	Possible							O			vel units	
606E <sub>h</sub>	-	VAR	Velocity window time	UNSIGNED16	rw	Possible							O			ms	
606F <sub>h</sub>	-	VAR	Velocity threshold	UNSIGNED16	rw	Possible							O			vel units	
6070 <sub>h</sub>	-	VAR	Velocity threshold time	UNSIGNED16	rw	Possible							O			ms	
60F <sub>h</sub>	-	VAR	Target velocity	INTEGER32	rw	Possible							M			vel units	
60F8 <sub>h</sub>	-	VAR	Max slippage	INTEGER32	ro	Possible							O			vel units	
60F9 <sub>h</sub>	-	ARRAY	Velocity control parameter set	UNSIGNED16	-	-							O			-	
	1	Gain	rw		Possible							M			-		
	2	T <sub>i</sub> - integration time constant	rw		Possible							M			-		
	3	manufacturer specific	manufacturer specific								O			-			
	...	.....	.....								...			-			
254			manufacturer specific		manufacturer specific								O			-	

## 19.1.9 Profile torque mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description	
							all	hm	pp	pc	ip	pv	tq	vl			
6071 <sub>h</sub>	-	VAR	Target torque	INTEGER16	rw	Possible							O	M		per thousand of rated torque	
6072 <sub>h</sub>	-	VAR	Max torque	UNSIGNED16	rw	Possible				O			O	O		per thousand of rated torque	
6073 <sub>h</sub>	-	VAR	Max current	UNSIGNED16	rw	Possible				O				O		per thousand of rated current	
6074 <sub>h</sub>	-	VAR	Torque demand value	INTEGER16	ro	Possible				O				O		per thousand of rated torque	
6075 <sub>h</sub>	-	VAR	Motor rated current	UNSIGNED32	rw	Possible				O				O		mA	
6076 <sub>h</sub>	-	VAR	Motor rated torque	UNSIGNED32	rw	Possible				O				O		0.001 Nm	
6077 <sub>h</sub>	-	VAR	Torque actual value	INTEGER16	ro	Possible				O				O		per thousand of rated torque	
6078 <sub>h</sub>	-	VAR	Current actual value	INTEGER16	ro	Possible				O				O		per thousand of rated current	
6079 <sub>h</sub>	-	VAR	DC link circuit voltage	UNSIGNED32	ro	Possible								O		mV	
6087 <sub>h</sub>	-	VAR	Torque slope	UNSIGNED32	rw	Possible								M		per thousand of rated torque per second	
6088 <sub>h</sub>	-	VAR	Torque profile type	INTEGER16	rw	Possible								M			
60F7 <sub>h</sub>	-	RECORD	Torque stage parameters	(manufacturer specific)	-	-								O		-	
	1		manufacturer specific		manufacturer specific										-		
	...		.....		.....										-		
			manufacturer specific		manufacturer specific										-		
	254				manufacturer specific										-		

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
60F6 <sub>h</sub>	-	RECORD	Torque control parameters	(manufacturer specific)	-	-	all	hm	pp	pc	ip	p <sub>v</sub>	tq	vl	-	
	1		manufacturer specific				manufacturer specific								-	
	...		.....				.....								-	
	...		manufacturer specific				manufacturer specific								-	
	254														-	

## 19.1.10 Velocity mode

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
6042 <sub>h</sub>	-	VAR	vl target velocity	INTEGER16	rw	Possible	all	hm	pp	pc	ip	p <sub>v</sub>	tq	vl	rpm	
6043 <sub>h</sub>	-	VAR	vl velocity demand	INTEGER16	ro	Possible									rpm	
6044 <sub>h</sub>	-	VAR	vl control effort	INTEGER16	ro	Possible									rpm	
6045 <sub>h</sub>	-	VAR	vl manipulated velocity	INTEGER16	ro	Possible									rpm	
6046 <sub>h</sub>	-	ARRAY	vl velocity min max amount	UNSIGNED32	-	-									M	-
	1		vl velocity min amount		rw	Possible									M	-
	2		vl velocity max amount		rw	Possible									M	-
6047 <sub>h</sub>	-	ARRAY	vl velocity min max	UNSIGNED32	-	-									O	-
	1		vl velocity min pos		rw	Possible									M	-
	2		vl velocity max pos		rw	Possible									M	-
	3		vl velocity min neg		rw	Possible									M	-
	4		vl velocity max neg		rw	Possible									M	-
6048 <sub>h</sub>	-	RECORD	vl velocity acceleration	vl velocity acceleration deceleration record	rw	-									M	-
	1		Delta speed		rw	Possible									M	rpm
	2		Delta time		rw	Possible									M	s

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vi		
6049 <sub>h</sub>	-	RECORD	vi velocity deceleration	vi velocity acceleration deceleration record	rw	-								M	-	
	1		Delta speed	UNSIGNED23	rw	Possible								M	rpm	
	2		Delta time	UNSIGNED16	rw	Possible								M	s	
604A <sub>h</sub>	-	RECORD	vi velocity quick stop	vi velocity acceleration deceleration record	rw	-								O	-	
	1		Delta speed	UNSIGNED32	rw	Possible								M	rpm	
	2		Delta time	UNSIGNED16	rw	Possible								M	s	
604B <sub>h</sub>	-	ARRAY	vi set-point factor	INTEGER16	rw	-								O	-	
	1		vi set-point factor numerator		rw	Possible								M	-	
	2		vi set-point factor denominator		rw	Possible								M	-	
604C <sub>h</sub>	-	ARRAY	vi dimension factor	INTEGER32	-	-								O	-	
	1		vi dimension factor numerator		rw	Possible								M	-	
	2		vi dimension factor denominator		rw	Possible								M	-	
604D <sub>h</sub>	-	VAR	vi pole number	UNSIGNED8	rw	Possible								O	-	
604E <sub>h</sub>	-	VAR	vi velocity reference	UNSIGNED32	rw	Possible								O	rpm	
604F <sub>h</sub>	-	VAR	vi ramp function time	UNSIGNED32	rw	Possible								O	ms	
6050 <sub>h</sub>	-	VAR	vi slow down time	UNSIGNED32	rw	Possible								O	ms	
6051 <sub>h</sub>	-	VAR	vi quick stop time	UNSIGNED32	rw	Possible								O	ms	
6052 <sub>h</sub>	-	VAR	vi nominal percentage	INTEGER16	rw	Possible								O	-	
6053 <sub>h</sub>	-	VAR	vi percentage demand	INTEGER16	ro	Possible								O	-	
6054 <sub>h</sub>	-	VAR	vi actual percentage	INTEGER16	ro	Possible								O	-	
6055 <sub>h</sub>	-	VAR	vi manipulated percentage	INTEGER16	ro	Possible								O	-	
6056 <sub>h</sub>	-	ARRAY	vi velocity motor min max amount	UNSIGNED32	-	-								O	-	
	1		vi velocity motor min amount		rw	Possible								M	(1/1000) rpm	
	2		vi velocity motor max amount		rw	Possible								M	(1/1000) rpm	



Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional							Units	Description		
							all	hnm	pp	pc	ip	pV	tq			vI	
6057 <sub>h</sub>	-	ARRAY	vI velocity motor min max	UNSIGNED32	-	-								O	-		
	1		vI velocity motor min pos		rw	Possible								M	(1/1000) rpm		
	2		vI velocity motor max pos		rw	Possible								M	(1/1000) rpm		
	3		vI velocity motor min neg		rw	Possible								M	(1/1000) rpm		
6058 <sub>h</sub>	4		vI velocity motor max neg		rw	Possible								M	(1/1000) rpm		
	-		ARRAY		vI frequency motor min max amount	-	-								O	-	
	1				vI frequency motor min amount	rw	Possible								M	-	
	2				vI frequency motor max amount	rw	Possible								M	-	
6059 <sub>h</sub>	-	ARRAY	vI frequency motor min max	UNSIGNED32	-	-								O	-		
	1		vI frequency motor min pos		rw	Possible								M	-		
	2		vI frequency motor max pos		rw	Possible								M	-		
	3		vI frequency motor min neg		rw	Possible								M	-		
	4		vI frequency motor max neg		rw	Possible								M	-		

## 19.2 Object dictionary by index

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
6007 <sub>h</sub>	-	VAR	Abort connection option code	INTEGER16	rw	Possible	O								-	
603F <sub>h</sub>	-	VAR	Error code	UNSIGNED16	ro	Possible	O								-	
6040 <sub>h</sub>	-	VAR	Controlword	UNSIGNED16	rw	Possible	M								-	
6041 <sub>h</sub>	-	VAR	Statusword	UNSIGNED16	ro	Possible	M								-	
6042 <sub>h</sub>	-	VAR	VI target velocity	INTEGER16	rw	Possible									rpm	
6043 <sub>h</sub>	-	VAR	VI velocity demand	INTEGER16	ro	Possible									rpm	
6044 <sub>h</sub>	-	VAR	VI control effort	INTEGER16	ro	Possible									rpm	
6045 <sub>h</sub>	-	VAR	VI manipulated velocity	INTEGER16	ro	Possible									rpm	
6046 <sub>h</sub>	-	ARRAY	VI velocity min max amount	UNSIGNED32	rw	-									-	
6047 <sub>h</sub>	-	ARRAY	VI velocity min max	UNSIGNED32	rw	-									-	
6048 <sub>h</sub>	-	RECORD	VI velocity acceleration	VI velocity acceleration deceleration record	rw	-									-	
6049 <sub>h</sub>	-	RECORD	VI velocity deceleration	VI velocity acceleration deceleration record	rw	-									-	
604A <sub>h</sub>	-	RECORD	VI velocity quick stop	VI velocity acceleration deceleration record	rw	-									-	
604B <sub>h</sub>	-	ARRAY	VI set-point factor	INTEGER16	rw	-									-	
604C <sub>h</sub>	-	ARRAY	VI dimension factor	INTEGER32	rw	-									-	
604D <sub>h</sub>	-	VAR	VI pole number	UNSIGNED8	rw	Possible									-	
604E <sub>h</sub>	-	VAR	VI velocity reference	UNSIGNED32	rw	Possible									rpm	
604F <sub>h</sub>	-	VAR	VI ramp function time	UNSIGNED32	rw	Possible									ms	
6050 <sub>h</sub>	-	VAR	VI slow down time	UNSIGNED32	rw	Possible									ms	
6051 <sub>h</sub>	-	VAR	VI quick stop time	UNSIGNED32	rw	Possible									ms	
6052 <sub>h</sub>	-	VAR	VI nominal percentage	INTEGER16	rw	Possible									-	
6053 <sub>h</sub>	-	VAR	VI percentage demand	INTEGER16	ro	Possible									-	
6054 <sub>h</sub>	-	VAR	VI actual percentage	INTEGER16	ro	Possible									-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
6055 <sub>h</sub>	-	VAR	vl manipulated percentage	INTEGER16	ro	Possible								O	-	
6056 <sub>h</sub>	-	ARRAY	vl velocity motor min max amount	UNSIGNED32	rw	-								O	(1/1000) rpm	
6057 <sub>h</sub>	-	ARRAY	vl velocity motor min max	UNSIGNED32	rw	-								O	(1/1000) rpm	
6058 <sub>h</sub>	-	ARRAY	vl frequency motor min max amount	UNSIGNED32	rw	-								O	-	
6059 <sub>h</sub>	-	ARRAY	vl frequency motor min max	UNSIGNED32	rw	-								O	-	
605A <sub>h</sub>	-	VAR	Quick stop option code	INTEGER16	rw	No	O								-	
605B <sub>h</sub>	-	VAR	Shutdown option code	INTEGER16	rw	No	O								-	
605C <sub>h</sub>	-	VAR	Disable operation option code	INTEGER16	rw	No	O								-	
605D <sub>h</sub>	-	VAR	Halt option code	INTEGER16	rw	No	O								-	
605E <sub>h</sub>	-	VAR	Fault reaction option code	INTEGER16	rw	No	O								-	
6060 <sub>h</sub>	-	VAR	Modes of operation	INTEGER8	rw	Possible	M								-	
6061 <sub>h</sub>	-	VAR	Modes of operation display	INTEGER8	ro	Possible	M								-	
6062 <sub>h</sub>	-	VAR	Position demand value	INTEGER32	ro	Possible				O					pos units	
6063 <sub>h</sub>	-	VAR	Position actual value*	INTEGER32	ro	Possible				O		O			inc	
6064 <sub>h</sub>	-	VAR	Position actual value	INTEGER32	ro	Possible		O	O	M	O		O		pos units	
6065 <sub>h</sub>	-	VAR	Following error window	UNSIGNED32	rw	Possible				O					pos units	
6066 <sub>h</sub>	-	VAR	Following error time out	UNSIGNED16	rw	Possible				O					ms	
6067 <sub>h</sub>	-	VAR	Position window	UNSIGNED32	rw	Possible				O					pos units	
6068 <sub>h</sub>	-	VAR	Position window time	UNSIGNED16	rw	Possible				O					ms	
6069 <sub>h</sub>	-	VAR	Velocity sensor actual value	INTEGER32	ro	Possible						M			inc / s	
606A <sub>h</sub>	-	VAR	Sensor selection code	INTEGER16	rw	Possible						O			-	
606B <sub>h</sub>	-	VAR	Velocity demand value	INTEGER32	ro	Possible						M			vel units	
606C <sub>h</sub>	-	VAR	Velocity actual value	INTEGER32	ro	Possible						M			vel units	
606D <sub>h</sub>	-	VAR	Velocity window	UNSIGNED16	rw	Possible						O			vel units	
606E <sub>h</sub>	-	VAR	Velocity window time	UNSIGNED16	rw	Possible						O			ms	
606F <sub>h</sub>	-	VAR	Velocity threshold	UNSIGNED16	rw	Possible						O			vel units	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description	
							all	hm	pp	pc	ip	pV	tq	VI			
6070 <sub>h</sub>	-	VAR	Velocity threshold time	UNSIGNED16	r/w	Possible							O			ms	
6071 <sub>h</sub>	-	VAR	Target torque	INTEGER16	rw	Possible							O	M		per thousand of rated torque	
6072 <sub>h</sub>	-	VAR	Max torque	UNSIGNED16	rw	Possible							O	O		per thousand of rated torque	
6073 <sub>h</sub>	-	VAR	Max current	UNSIGNED16	rw	Possible							O			per thousand of rated current	
6074 <sub>h</sub>	-	VAR	Torque demand value	INTEGER16	ro	Possible								O		per thousand of rated torque	
6075 <sub>h</sub>	-	VAR	Motor rated current	UNSIGNED32	rw	Possible								O		mA	
6076 <sub>h</sub>	-	VAR	Motor rated torque	UNSIGNED32	rw	Possible								O		0.001 Nm	
6077 <sub>h</sub>	-	VAR	Torque actual value	INTEGER16	ro	Possible								O		per thousand of rated torque	
6078 <sub>h</sub>	-	VAR	Current actual value	INTEGER16	ro	Possible								O		per thousand of rated current	
6079 <sub>h</sub>	-	VAR	DC link circuit voltage	UNSIGNED32	ro	Possible									O	mV	
607A <sub>h</sub>	-	VAR	Target position	INTEGER32	rw	Possible			M	M						pos units	
607B <sub>h</sub>	-	ARRAY	Position range limit	INTEGER32	rw	-			O	O						-	
607C <sub>h</sub>	-	VAR	Home offset	INTEGER32	rw	Possible		O		O						pos units	
607D <sub>h</sub>	-	ARRAY	Software position limit	INTEGER32	rw	-			O	O						pos units	
607E <sub>h</sub>	-	VAR	Polarity	UNSIGNED8	rw	Possible			O	O	O	O	O	O		-	
607F <sub>h</sub>	-	VAR	Max profile velocity	UNSIGNED32	rw	Possible			O			O				speed units	
6080 <sub>h</sub>	-	VAR	Max motor speed	UNSIGNED32	rw	Possible			O			O				rpm	
6081 <sub>h</sub>	-	VAR	Profile velocity	UNSIGNED32	rw	Possible			M			M				speed units	
6082 <sub>h</sub>	-	VAR	End velocity	UNSIGNED32	rw	Possible			O							speed units	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vI		
6083 <sub>h</sub>	-	VAR	Profile acceleration	UNSIGNED32	rw	Possible			M			M			acc units	
6084 <sub>h</sub>	-	VAR	Profile deceleration	UNSIGNED32	rw	Possible			M			M			acc units	
6085 <sub>h</sub>	-	VAR	Quick stop deceleration	UNSIGNED32	rw	Possible			O			O			acc units	
6086 <sub>h</sub>	-	VAR	Motion profile type	INTEGER16	rw	Possible			M			M			-	
6087 <sub>h</sub>	-	VAR	Torque slope	UNSIGNED32	rw	Possible							M		per thousand of rated torque per second	
6088 <sub>h</sub>	-	VAR	Torque profile type	INTEGER16	rw	Possible							M		-	
6089 <sub>h</sub>	-	VAR	Position notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608A <sub>h</sub>	-	VAR	Position dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608B <sub>h</sub>	-	VAR	Velocity notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608C <sub>h</sub>	-	VAR	Velocity dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608D <sub>h</sub>	-	VAR	Acceleration notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
608E <sub>h</sub>	-	VAR	Acceleration dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
608F <sub>h</sub>	-	ARRAY	Position encoder resolution	UNSIGNED32	rw	-			O		O	O	O		inc / rev	
6090 <sub>h</sub>	-	ARRAY	Velocity encoder resolution	UNSIGNED32	rw	-			O		O	O	O		inc / rev	
6091 <sub>h</sub>	-	ARRAY	Gear ratio	UNSIGNED32	rw	-			O		O	O	O		-	
6092 <sub>h</sub>	-	ARRAY	Feed constant	UNSIGNED32	rw	-			O		O	O	O		pos units / rev	
6093 <sub>h</sub>	-	ARRAY	Position factor	UNSIGNED32	rw	-			O	O	O				-	
6094 <sub>h</sub>	-	ARRAY	Velocity encoder factor	UNSIGNED32	rw	-			O	O		O			-	
6095 <sub>h</sub>	-	ARRAY	Velocity factor 1	UNSIGNED32	rw	-			O	O	O	O	O		-	
6096 <sub>h</sub>	-	ARRAY	Velocity factor 2	UNSIGNED32	rw	-			O		O	O	O		-	
6097 <sub>h</sub>	-	ARRAY	Acceleration factor	UNSIGNED32	rw	-			O	O	O	O	O		-	
6098 <sub>h</sub>	-	VAR	Homing method	INTEGER8	rw	Possible		M							-	
6099 <sub>h</sub>	-	ARRAY	Homing speeds	UNSIGNED32	rw	-		M							vel units	
609A <sub>h</sub>	-	VAR	Homing acceleration	UNSIGNED32	rw	Possible		O							acc units	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
60C0 <sub>h</sub>	-	VAR	Interpolation sub mode select	INTEGER16	rw	Possible					O				-	
60C1 <sub>h</sub>	-	ARRAY	Interpolation data record	(see 60C0 <sub>h</sub> )	rw	-					O				-	
60C2 <sub>h</sub>	-	RECORD	Interpolation time period	Interpolation time period record	rw	-					O				-	
60C3 <sub>h</sub>	-	ARRAY	Interpolation sync definition	UNSIGNED8	rw	-					O				-	
60C4 <sub>h</sub>	-	RECORD	Interpolation data configuration	Interpolation data configuration record	-	-					O				-	
60C5 <sub>h</sub>	-	VAR	Max acceleration	UNSIGNED32	rw	Possible			O						acc units	
60C6 <sub>h</sub>	-	VAR	Max deceleration	UNSIGNED32	rw	Possible			O						acc units	
60F4 <sub>h</sub>	-	VAR	Following error actual value	INTEGER32	ro	Possible				O					pos units	
60F6 <sub>h</sub>	-	RECORD	Torque control parameters	(manufacturer specific)	-	-							O		-	
60F7 <sub>h</sub>	-	RECORD	Power stage parameters	(manufacturer specific)	-	-						O			-	
60F8 <sub>h</sub>	-	VAR	Max slippage	INTEGER32	ro	Possible					O				vel units	
60F9 <sub>h</sub>	-	ARRAY	Velocity control parameter set	UNSIGNED16	rw	-						O			-	
60FA <sub>h</sub>	-	VAR	Control effort	INTEGER32	ro	Possible				O					-	
60FB <sub>h</sub>	-	RECORD	Position control parameter set	(manufacturer specific)	-	-				O					-	
60FC <sub>h</sub>	-	VAR	Position demand value*	INTEGER32	ro	Possible				O					inc	
60FD <sub>h</sub>	-	VAR	Digital inputs	UNSIGNED32	ro	Possible	O								-	
60FE <sub>h</sub>	-	ARRAY	Digital outputs	UNSIGNED32	rw	-	O								-	
60FF <sub>h</sub>	-	VAR	Target velocity	INTEGER32	rw	Possible						M			vel units	
6402 <sub>h</sub>	-	VAR	Motor type	UNSIGNED16	rw	Possible	O								-	
6403 <sub>h</sub>	-	VAR	Motor catalog number	VISIBLE_STRING	rw	No	O								-	
6404 <sub>h</sub>	-	VAR	Motor manufacturer	VISIBLE_STRING	rw	No	O								-	
6405 <sub>h</sub>	-	VAR	http motor catalog address	VISIBLE_STRING	rw	No	O								-	
6406 <sub>h</sub>	-	VAR	Motor calibration date	TIME_OF_DAY	rw	Possible	O								-	
6407 <sub>h</sub>	-	VAR	Motor service period	UNSIGNED32	rw	Possible	O								-	
6410 <sub>h</sub>	-	RECORD	Motor data	(manufacturer specific)	rw	-	O								-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	VI		
6502 <sub>h</sub>	-	VAR	Supported drive modes	UNSIGNED32	ro	Possible	O								-	
6503 <sub>h</sub>	-	VAR	Drive catalog number	VISIBLE_STRING	rw	No	O								-	
6504 <sub>h</sub>	-	VAR	Drive manufacturer	VISIBLE_STRING	rw	No	O								-	
6505 <sub>h</sub>	-	VAR	http drive catalog address	VISIBLE_STRING	rw	No	O								-	
6510 <sub>h</sub>	-	RECORD	Drive data	(manufacturer specific)	-	-	O								-	

## 19.3 Object dictionary by name

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	VI		
6007 <sub>h</sub>	-	VAR	Abort connection option code	INTEGER16	rw	Possible	O								-	
608E <sub>h</sub>	-	VAR	Acceleration dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
6097 <sub>h</sub>	-	ARRAY	Acceleration factor	UNSIGNED32	rw	-			O	O	O	O	O		-	
608D <sub>h</sub>	-	VAR	Acceleration notation index	INTEGER8	rw	Possible		C	C		C	C	C		-	
60FA <sub>h</sub>	-	VAR	Control effort	INTEGER32	ro	Possible				O					-	
6040 <sub>h</sub>	-	VAR	Controlword	UNSIGNED16	rw	Possible	M								-	
6078 <sub>h</sub>	-	VAR	Current actual value	INTEGER16	ro	Possible							O		per thousand of rated current	
6079 <sub>h</sub>	-	VAR	DC link circuit voltage	UNSIGNED32	ro	Possible							O		mV	
60FD <sub>h</sub>	-	VAR	Digital inputs	UNSIGNED32	ro	Possible	O								-	
60FE <sub>h</sub>	-	ARRAY	Digital outputs	UNSIGNED32	rw	-	O								-	
605C <sub>h</sub>	-	VAR	Disable operation option code	INTEGER16	rw	No	O								-	
6503 <sub>h</sub>	-	VAR	Drive catalog number	VISIBLE_STRING	rw	No	O								-	
6510 <sub>h</sub>	-	RECORD	Drive data	(manufacturer specific)	-	-	O								-	
6504 <sub>h</sub>	-	VAR	Drive manufacturer	VISIBLE_STRING	rw	No	O								-	
6082 <sub>h</sub>	-	VAR	End velocity	UNSIGNED32	rw	Possible			O						speed units	
603F <sub>h</sub>	-	VAR	Error code	UNSIGNED16	ro	Possible	O								-	
605E <sub>h</sub>	-	VAR	Fault reaction option code	INTEGER16	rw	No	O								-	
6092 <sub>h</sub>	-	ARRAY	Feed constant	UNSIGNED32	rw	-			O		O	O	O		pos units / rev	
60F4 <sub>h</sub>	-	VAR	Following error actual value	INTEGER32	ro	Possible				O					pos units	
6066 <sub>h</sub>	-	VAR	Following error time out	UNSIGNED16	rw	Possible				O					ms	
6065 <sub>h</sub>	-	VAR	Following error window	UNSIGNED32	rw	Possible				O					pos units	
6091 <sub>h</sub>	-	ARRAY	Gear ratio	UNSIGNED32	rw	-			O		O	O	O		-	
605D <sub>h</sub>	-	VAR	Halt option code	INTEGER16	rw	No	O								-	



Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vI		
607C <sub>h</sub>	-	VAR	Home offset	INTEGER32	rw	Possible		O		O					pos units	
609A <sub>h</sub>	-	VAR	Homing acceleration	UNSIGNED32	rw	Possible		O							acc units	
6098 <sub>h</sub>	-	VAR	Homing method	INTEGER8	rw	Possible		M							-	
6099 <sub>h</sub>	-	ARRAY	Homing speeds	UNSIGNED32	rw	-		M							vel units	
6505 <sub>h</sub>	-	VAR	http drive catalog address	VISIBLE_STRING	rw	No	O								-	
6405 <sub>h</sub>	-	VAR	http motor catalog address	VISIBLE_STRING	rw	No	O								-	
60C4 <sub>h</sub>	-	RECORD	Interpolation data configuration	Interpolation data configuration record	-	-				O					-	
60C1 <sub>h</sub>	-	ARRAY	Interpolation data record	(see 60C0 <sub>h</sub> )	rw	-				O					-	
60C0 <sub>h</sub>	-	VAR	Interpolation sub mode select	INTEGER16	rw	Possible				O					-	
60C3 <sub>h</sub>	-	ARRAY	Interpolation sync definition	UNSIGNED8	rw	-				O					-	
60C2 <sub>h</sub>	-	RECORD	Interpolation time period	Interpolation time period record	rw	-				O					-	
60C5 <sub>h</sub>	-	VAR	Max acceleration	UNSIGNED32	rw	Possible		O							acc units	
6073 <sub>h</sub>	-	VAR	Max current	UNSIGNED16	rw	Possible						O			per thousand of rated current	
60C6 <sub>h</sub>	-	VAR	Max deceleration	UNSIGNED32	rw	Possible			O						acc units	
6080 <sub>h</sub>	-	VAR	Max motor speed	UNSIGNED32	rw	Possible			O		O				rpm	
607F <sub>h</sub>	-	VAR	Max profile velocity	UNSIGNED32	rw	Possible			O		O				speed units	
60F8 <sub>h</sub>	-	VAR	Max slippage	INTEGER32	ro	Possible					O				vel units	
6072 <sub>h</sub>	-	VAR	Max torque	UNSIGNED16	rw	Possible					O	O			per thousand of rated torque	
6060 <sub>h</sub>	-	VAR	Modes of operation	INTEGER8	rw	Possible	M								-	
6061 <sub>h</sub>	-	VAR	Modes of operation display	INTEGER8	ro	Possible	M								-	
6086 <sub>h</sub>	-	VAR	Motion profile type	INTEGER16	rw	Possible			M		M				-	
6406 <sub>h</sub>	-	VAR	Motor calibration date	TIME_OF_DAY	rw	Possible	O								-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vl		
6403 <sub>h</sub>	-	VAR	Motor catalog number	VISIBLE_STRING	rw	No	O								-	
6410 <sub>h</sub>	-	RECORD	Motor data	(manufacturer specific)	rw	-	O								-	
6404 <sub>h</sub>	-	VAR	Motor manufacturer	VISIBLE_STRING	rw	No	O								-	
6075 <sub>h</sub>	-	VAR	Motor rated current	UNSIGNED32	rw	Possible							O		mA	
6076 <sub>h</sub>	-	VAR	Motor rated torque	UNSIGNED32	rw	Possible							O		0.001 Nm	
6407 <sub>h</sub>	-	VAR	Motor service period	UNSIGNED32	rw	Possible	O								-	
6402 <sub>h</sub>	-	VAR	Motor type	UNSIGNED16	rw	Possible	O								-	
607E <sub>h</sub>	-	VAR	Polarity	UNSIGNED8	rw	Possible			O	O	O	O	O		-	
6064 <sub>h</sub>	-	VAR	Position actual value	INTEGER32	ro	Possible		O	O	M	O		O		pos units	
6063 <sub>h</sub>	-	VAR	Position actual value *	INTEGER32	ro	Possible				O		O			inc	
60FB <sub>h</sub>	-	RECORD	Position control parameter set	(manufacturer specific)	-	-				O					-	
6062 <sub>h</sub>	-	VAR	Position demand value	INTEGER32	ro	Possible				O					pos units	
60FC <sub>h</sub>	-	VAR	Position demand value*	INTEGER32	ro	Possible					O				inc	
608A <sub>h</sub>	-	VAR	Position dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C	C	-	
608F <sub>h</sub>	-	ARRAY	Position encoder resolution	UNSIGNED32	rw	-			O		O	O	O	O	inc / rev	
6093 <sub>h</sub>	-	ARRAY	Position factor	UNSIGNED32	rw	-				O	O	O			-	
6089 <sub>h</sub>	-	VAR	Position notation index	INTEGER8	rw	Possible		C	C		C	C	C	C	-	
607B <sub>h</sub>	-	ARRAY	Position range limit	INTEGER32	rw	-			O	O					-	
6067 <sub>h</sub>	-	VAR	Position window	UNSIGNED32	rw	Possible				O					pos units	
6068 <sub>h</sub>	-	VAR	Position window time	UNSIGNED16	rw	Possible				O					ms	
60F7 <sub>h</sub>	-	RECORD	Power stage parameters	(manufacturer specific)	-	-							O		-	
6083 <sub>h</sub>	-	VAR	Profile acceleration	UNSIGNED32	rw	Possible			M			M			acc units	
6084 <sub>h</sub>	-	VAR	Profile deceleration	UNSIGNED32	rw	Possible			M			M			acc units	
6081 <sub>h</sub>	-	VAR	Profile velocity	UNSIGNED32	rw	Possible			M			M			speed units	
6085 <sub>h</sub>	-	VAR	Quick stop deceleration	UNSIGNED32	rw	Possible			O			O			acc units	
605A <sub>h</sub>	-	VAR	Quick stop option code	INTEGER16	rw	No	O								-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	vI		
606A <sub>h</sub>	-	VAR	Sensor selection code	INTEGER16	rw	Possible						O			-	
605B <sub>h</sub>	-	VAR	Shutdown option code	INTEGER16	rw	No	O								-	
607D <sub>h</sub>	-	ARRAY	Software position limit	INTEGER32	rw	-			O	O					pos units	
6041 <sub>h</sub>	-	VAR	Statusword	UNSIGNED16	ro	Possible	M								-	
6502 <sub>h</sub>	-	VAR	Supported drive modes	UNSIGNED32	ro	Possible	O								-	
607A <sub>h</sub>	-	VAR	Target position	INTEGER32	rw	Possible			M	M					pos units	
6071 <sub>h</sub>	-	VAR	Target torque	INTEGER16	rw	Possible						O	M		per thousand of rated torque	
60FF <sub>h</sub>	-	VAR	Target velocity	INTEGER32	rw	Possible						M			vel units	
6077 <sub>h</sub>	-	VAR	Torque actual value	INTEGER16	ro	Possible							O		per thousand of rated torque	
60F6 <sub>h</sub>	-	RECORD	Torque control parameters	(manufacturer specific)	-	-							O		-	
6074 <sub>h</sub>	-	VAR	Torque demand value	INTEGER16	ro	Possible							O		per thousand of rated torque	
6088 <sub>h</sub>	-	VAR	Torque profile type	INTEGER16	rw	Possible							M		-	
6087 <sub>h</sub>	-	VAR	Torque slope	UNSIGNED32	rw	Possible							M		per thousand of rated torque per second	
606C <sub>h</sub>	-	VAR	Velocity actual value	INTEGER32	ro	Possible						M			vel units	
60F9 <sub>h</sub>	-	ARRAY	Velocity control parameter set	UNSIGNED16	rw	-						O			-	
606B <sub>h</sub>	-	VAR	Velocity demand value	INTEGER32	ro	Possible						M			vel units	
608C <sub>h</sub>	-	VAR	Velocity dimension index	UNSIGNED8	rw	Possible		C	C		C	C	C		-	
6094 <sub>h</sub>	-	ARRAY	Velocity encoder factor	UNSIGNED32	rw	-			O	O		O			-	
6090 <sub>h</sub>	-	ARRAY	Velocity encoder resolution	UNSIGNED32	rw	-			O		O	O	O		inc / rev	
6095 <sub>h</sub>	-	ARRAY	Velocity factor 1	UNSIGNED32	rw	-			O	O	O	O	O		-	
6096 <sub>h</sub>	-	ARRAY	Velocity factor 2	UNSIGNED32	rw	-			O		O	O	O		-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description	
							all	Mandatory			Optional			Conditional			
								hm	pp	pc	ip	pV	tq	vI			
608B <sub>h</sub>	-	VAR	Velocity notation index	INTEGER8	rw	Possible			C	C			C	C		-	
6069 <sub>h</sub>	-	VAR	Velocity sensor actual value	INTEGER32	ro	Possible							M			inc / s	
606F <sub>h</sub>	-	VAR	Velocity threshold	UNSIGNED16	rw	Possible							O			vel units	
6070 <sub>h</sub>	-	VAR	Velocity threshold time	UNSIGNED16	r/w	Possible							O			ms	
606D <sub>h</sub>	-	VAR	Velocity window	UNSIGNED16	rw	Possible							O			vel units	
606E <sub>h</sub>	-	VAR	Velocity window time	UNSIGNED16	rw	Possible						O				ms	
6054 <sub>h</sub>	-	VAR	vl actual percentage	INTEGER16	ro	Possible									O	-	
6044 <sub>h</sub>	-	VAR	vl control effort	INTEGER16	ro	Possible									M	rpm	
604C <sub>h</sub>	-	ARRAY	vl dimension factor	INTEGER32	rw	-									O	-	
6059 <sub>h</sub>	-	ARRAY	vl frequency motor min max	UNSIGNED32	rw	-									O	-	
6058 <sub>h</sub>	-	ARRAY	vl frequency motor min max amount	UNSIGNED32	rw	-									O	-	
6055 <sub>h</sub>	-	VAR	vl manipulated percentage	INTEGER16	ro	Possible									O	-	
6045 <sub>h</sub>	-	VAR	vl manipulated velocity	INTEGER16	ro	Possible									O	rpm	
6052 <sub>h</sub>	-	VAR	vl nominal percentage	INTEGER16	rw	Possible									O	-	
6053 <sub>h</sub>	-	VAR	vl percentage demand	INTEGER16	ro	Possible									O	-	
604D <sub>h</sub>	-	VAR	vl pole number	UNSIGNED8	rw	Possible									O	-	
6051 <sub>h</sub>	-	VAR	vl quick stop time	UNSIGNED32	rw	Possible									O	ms	
604F <sub>h</sub>	-	VAR	vl ramp function time	UNSIGNED32	rw	Possible									O	ms	
604B <sub>h</sub>	-	ARRAY	vl set-point factor	INTEGER16	rw	-									O	-	
6050 <sub>h</sub>	-	VAR	vl slow down time	UNSIGNED32	rw	Possible									O	ms	
6042 <sub>h</sub>	-	VAR	vl target velocity	INTEGER16	rw	Possible									M	rpm	
6048 <sub>h</sub>	-	RECORD	vl velocity acceleration	vl velocity acceleration deceleration record	rw	-									M	-	
6049 <sub>h</sub>	-	RECORD	vl velocity deceleration	vl velocity acceleration deceleration record	rw	-									M	-	
6043 <sub>h</sub>	-	VAR	vl velocity demand	INTEGER16	ro	Possible									M	rpm	
6047 <sub>h</sub>	-	ARRAY	vl velocity min max	UNSIGNED32	rw	-									O	-	

Index	Sub	Object	Name	Type	Attr.	PDO Mapping	Mandatory / Optional / Conditional								Units	Description
							all	hm	pp	pc	ip	pV	tq	VI		
6046 <sub>h</sub>	-	ARRAY	VI velocity min max amount	UNSIGNED32	rw	-								M	-	
6057 <sub>h</sub>	-	ARRAY	VI velocity motor min max	UNSIGNED32	rw	-								O	(1/1000) rpm	
6056 <sub>h</sub>	-	ARRAY	VI velocity motor min max amount	UNSIGNED32	rw	-								O	(1/1000) rpm	
604A <sub>h</sub>	-	RECORD	VI velocity quick stop	VI velocity acceleration deceleration record	rw	-								O	-	
604E <sub>h</sub>	-	VAR	VI velocity reference	UNSIGNED32	rw	Possible								O	rpm	

## 19.4 Definition of dimension indices

## 19.4.1 Dimension index table

Physical dimension	Unit	Dimension index
None	-	00 <sub>h</sub>
Way / length	m	01
Area	m <sup>2</sup>	A0 <sub>h</sub>
Volume	m <sup>3</sup>	A1 <sub>h</sub>
Time	s	A2 <sub>h</sub>
	min	47 <sub>h</sub>
	h	48 <sub>h</sub>
	d	49 <sub>h</sub>
	y	4A <sub>h</sub>
Power	W	24 <sub>h</sub>
Revolutions / time	rev / s	A3 <sub>h</sub>
	rev / min	A4 <sub>h</sub>
	rev / h	A5 <sub>h</sub>
Angle	rad	10 <sub>h</sub>
	s	43 <sub>h</sub>
	m	42 <sub>h</sub>
	°	41 <sub>h</sub>
Velocity	m / s	A6 <sub>h</sub>
	m / min	A7 <sub>h</sub>
	m / h	A8 <sub>h</sub>
Torque	N / m	A9 <sub>h</sub>
Temperature	K	05 <sub>h</sub>
	°C	2D <sub>h</sub>
	F	Aa <sub>h</sub>
Voltage	V	26 <sub>h</sub>
Current	A	04 <sub>h</sub>
Ratio	%	Ab <sub>h</sub>
Frequency	Hz	20 <sub>h</sub>
Steps	steps	Ac <sub>h</sub>
Steps / revolution	steps / rev	Ad <sub>h</sub>

## 19.4.2 Notation index table

Prefix	Factor	Symbol	Notation index
reserved	-	-	13h ... 7F <sub>h</sub>
exa	10 <sup>18</sup>	E	12 <sub>h</sub>
-	10 <sup>17</sup>	-	11 <sub>h</sub>
-	10 <sup>16</sup>	-	10 <sub>h</sub>
peta	10 <sup>15</sup>	P	0F <sub>h</sub>
-	10 <sup>14</sup>	-	0E <sub>h</sub>
-	10 <sup>13</sup>	-	0D <sub>h</sub>
tera	10 <sup>12</sup>	T	0C <sub>h</sub>
-	10 <sup>11</sup>	-	0B <sub>h</sub>
-	10 <sup>10</sup>	-	0A <sub>h</sub>
giga	10 <sup>9</sup>	G	09 <sub>h</sub>
-	10 <sup>8</sup>	-	08 <sub>h</sub>
-	10 <sup>7</sup>	-	07 <sub>h</sub>
mega	10 <sup>6</sup>	M	06 <sub>h</sub>
-	10 <sup>5</sup>	-	05 <sub>h</sub>
-	10 <sup>4</sup>	-	04 <sub>h</sub>
kilo	10 <sup>3</sup>	k	03 <sub>h</sub>
hecto	10 <sup>2</sup>	h	02 <sub>h</sub>
deca	10 <sup>1</sup>	da	01 <sub>h</sub>
-	10 <sup>0</sup>	-	00 <sub>h</sub>
deci	10 <sup>-1</sup>	d	FF <sub>h</sub>
centi	10 <sup>-2</sup>	c	FE <sub>h</sub>
milli	10 <sup>-3</sup>	m	FD <sub>h</sub>
-	10 <sup>-4</sup>	-	FC <sub>h</sub>
-	10 <sup>-5</sup>	-	FB <sub>h</sub>
micro	10 <sup>-6</sup>	μ	FA <sub>h</sub>
-	10 <sup>-7</sup>	-	F9 <sub>h</sub>
-	10 <sup>-8</sup>	-	F8 <sub>h</sub>
nano	10 <sup>-9</sup>	n	F7 <sub>h</sub>
-	10 <sup>-10</sup>	-	F6 <sub>h</sub>
-	10 <sup>-11</sup>	-	F5 <sub>h</sub>
pico	10 <sup>-12</sup>	p	F4 <sub>h</sub>
-	10 <sup>-13</sup>	-	F3 <sub>h</sub>
-	10 <sup>-14</sup>	-	F2 <sub>h</sub>
femto	10 <sup>-15</sup>	f	F1 <sub>h</sub>
-	10 <sup>-16</sup>	-	F0 <sub>h</sub>
-	10 <sup>-17</sup>	-	EF <sub>h</sub>
atto	10 <sup>-18</sup>	a	EE <sub>h</sub>
reserved	-	-	ED <sub>h</sub> ... 80 <sub>h</sub>