CiA Draft Standard Proposal 402

Drives and motion control device profile

Part 2: Operation modes and application data

This DSP is for CiA members only and may be changed without notification.

Version: 3.0

14 December 2007

! CAN in Automation (CiA) e. V.

HISTORY

The CiA 402 CANopen device profile for drives and motion control was submitted to IEC for international standardization. The CiA 402 related clauses in the IEC 61800-7-1, IEC 61800-7-201 and IEC 61800-7-301 standards are published in this set of profile specifications. This set substitutes the CiA DSP 402 version 2.0 from July 2002. The device profile has been entirely reviewed and restructured. There have been dropped several object dictionary entries in order to simplify the implementation of this profile. On the other hand new operation modes have been introduced.

EDITORIAL REMARK

Part 1 of this set of specifications provides general information and presents the annex A of IEC 61800-7-1. Part 2 presents the IEC 61800-7-201 standard, and part 3 presents the CANopen-relevant clauses in IEC 61800-7-301.

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.

Because this specification is licensed free of charge, there is no warranty for this specification, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holder and/or other parties provide this specification “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the correctness and completeness of the specification is with you. Should this specification prove failures, you assume the cost of all necessary servicing, repair or correction.

Trademarks

CANopen® and CiA® are registered community trademarks of CAN in Automation. The use is restricted for CiA members or owners of CANopen vendor ID. More detailed terms for the use are available from CiA.

CAN in Automation e. V.
Kontumazgarten 3
DE - 90429 Nuremberg, Germany
Tel.: +49-911-928819-0
Fax: +49-911-928819-79
Url: www.can-cia.org
Email: headquarters@can-cia.org

CONTENTS

1 Scope

This specification specifies the CANopen device profile for drives and motion control. It consists of three parts:

- Part 1: General definitions
- Part 2: Operation modes and application data
- Part 3: PDO mapping

Part 2 specifies the basic drive and additional application functional elements as defined in IEC 61800-7-201. This includes the specification of drive operation modes and application data for power drive systems like frequency converters, servo controllers, or stepper motor controllers. It includes the definition of real-time control objects as well as of configuration, adjustment, identification and network management objects. There is also defined the PDS finite state automaton (FSA), which may be controlled externally by a control device communicating via a communication system to the drive device.

2 Normative references

/CiA301/ CiA 301, CANopen application layer and communication profile

/CiA402-3/ CiA 402-3, CANopen drive and motion control profile – Part 3: PDO mapping

/IEC61800-7-1/ IEC 61800-7-1, Adjustable speed electrical power drive systems – Generic interface and use of profiles for power drive systems – Interface definition

3 Abbreviations and definitions

3.1 Abbreviations

CAN Controller area network
COB Communication object
FSA Finite state automaton
ID Identifier
LSB Least significant bit
MSB Most significant bit
PDO Process data object
PDS Power drive system
RPDO Receive process data object
SDO Service data object
TPDO Transmit process data object

3.2 Definitions

The definitions given in /CiA301/ and /IEC61800-7-1/ apply for this specification, too.

4 Introduction

4.1 Operation modes

The device profile defines several modes of operation. They include profile position mode, homing mode, interpolated position mode, profile velocity mode, profile torque mode, velocity mode, cyclic synchronous position mode, cyclic synchronous velocity mode, and cyclic synchronous torque mode.

4.2 Object dictionary

All application objects described in this specification are grouped in the object dictionary, and defined by attributes as defined in /CiA301/. All objects shall be accessible via the network in an ordered pre-defined fashion by means of SDO. Each object within the dictionary shall be addressed uniquely by using a 16-bit index and an 8-bit sub-index. The communication-related objects are defined in /CiA301/ and /CiA402-3/.

The standardised device profile area at indices 6000_h through $9FFF_h$ shall contain all application objects common to this device profile specification. The following object indices shall be reserved for compatibility reasons: 6045_h, 6047_h, $604D_h$, $604E_h$, $604F_h$, 6052_h, 6053_h, 6054_h, 6055_h, 6056_h, 6057_h, 6058_h, 6059_h, 6089_h, $608A_h$, $608B_h$, $608C_h$, $608D_h$, $608E_h$, 6093_h, 6094_h, 6095_h, 6096_h, 6097_h, $60A0_h$, $60A1_h$, $60A2_h$, $60F6_h$, $60F7_h$, $60F9_h$, $60FB_h$, 6410_h, 6504_h, and 6510_h.

The objects may be read respectively written via the network. Within this range of objects up to 8 axes may be realised. Additionally it is possible to implement other device profiles (e. g. generic I/O module or encoder) within the CANopen drive device. These other CANopen device profiles may be implemented instead of one or several axes.

For multi axes devices the object range 6000_h to $67FF_h$ shall be shifted as follows:

- 6000_h to $67FF_h$: axis 0
- 6800_h to $6FFF_h$: axis 1
- 7000_h to $77FF_h$: axis 2
- 7800_h to $7FFF_h$: axis 3
- 8000_h to $87FF_h$: axis 4
- 8800_h to $8FFF_h$: axis 5
- 9000_h to $97FF_h$: axis 6
- 9800_h to $9FFF_h$: axis 7

The category and entry category attributes of an object indicate if the object shall be implemented (*Mandatory*) or may be implemented (*Optional*).

The object code and data type attributes are defined in detail in /CiA301/. The used data type attributes are given in Clause 5. In the entry description, the access attribute indicating if an application object is read only (*ro*), read/write (*rw*) or write only (*wo*) or constant (*c*) is defined. Read only indicates that this shall not be written via the bus; read/write allows to read and to write this object; and write only means that this application object shall be not read via the bus.

The PDO mapping attribute shall indicate if this object shall be or may be or shall not be mapped into PDO. The detailed definition of these attributes is given in /CiA402-3/.

The default value attribute defines the value of an object with access attribute of the value *'rw'* and *'c'* after power-on or application reset.

5 Data types

5.1 Standard data types

The data types used in this profile are listed in Table 1.

Table 1 — List of used data types

Data type	Reference
Unsigned8	/CiA301/
Unsigned16	/CiA301/
Unsigned32	/CiA301/
Integer8	/CiA301/
Integer16	/CiA301/
Integer32	/CiA301/
Visible string	/CiA301/
Time of day	/CiA301/
Interpolated time period	See Table 2
Interpolated data configuration	See Table 3
v/ velocity acceleration/deceleration	See Table 4

5.2 Record definitions

Table 2, Table 3, and Table 4 define the records used in this specification.

Table 2 — Interpolated time period

Index	Sub-index	Description	Data type
0080_h	00_h	Highest index supported	Unsigned8
	01_h	Interpolation time units	Unsigned8
	02_h	Interpolation time index	Integer8

Table 3 — Interpolated data configuration

Index	Sub-index	Description	Data type
0081_h	00_h	Highest index supported	Unsigned8
	01_h	Maximum buffer size	Unsigned32
	02_h	Actual buffer size	Unsigned32
	03_h	Buffer organisation	Unsigned8
	04_h	Buffer position	Unsigned16
	05_h	Size of data record	Unsigned8
	06_h	Buffer clear	Unsigned8

Table 4 — *vl* velocity acceleration/deceleration

Index	Sub-index	Description	Data type
0082_h	00_h	Highest index supported	Unsigned8
	01_h	Delta speed	Unsigned32
	02_h	Delta time	Integer16

6 General object definitions

6.1 General

In the following clauses, the communication parameter objects, the additional identification and the information objects are defined.

6.2 Communication parameter objects

There are three communication parameter objects that shall be implemented:

- Device type object 1000_h
- Error register object 1001_h
- Identity object 1018_h

They are defined in /CiA301/ and the following definitions shall also apply.

The device type object shall define the device type, the device's functionality, and the mapping variant.

For multi device modules, the additional information parameter shall contain $0FFF_h$ and the device profile number referenced by object 1000_h is the device profile of the first device in the object dictionary. All other devices of a multiple device module shall identify their profiles at object $67FF_h + x \times 800_h$ with x = internal number of the device (0 to 7). For details, see /CiA301/. Figure 1 specifies the structure and the values of the *device type* object, Table 5 specifies the object description, and Table 6 specifies the entry desciption.

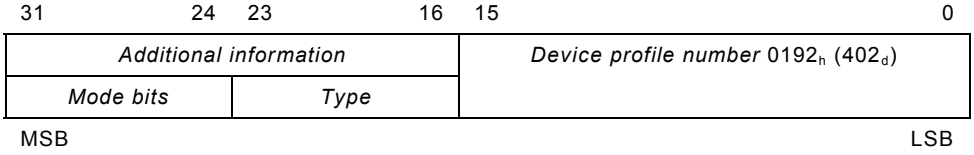


Figure 1 – Value definition

Mode bits and *type* in the *additional information* are defined in /CiA402-3/.

Table 5 – Object description

Attribute	Value
Index	1000_h
Name	Device type
Object code	See /CiA301/
Data type	See /CiA301/
Category	Mandatory

Table 6 – Entry description

Attribute	Value
Sub-Index	00_h
Access	c
PDO mapping	See /CiA402-3/
Value range	See value definition
Default value	Manufacturer-specific

The device-profile specific bit in the *error register* object (1001_h) shall be used to indicate that the *error code* in the Emergency message is defined in this part of profile specification.

NOTE The corresponding *error code* may be read in object 1003_h (see /CiA301/) or object $603F_h$.

6.3 Additional identification and information objects

6.3.1 Object 6402_h: Motor type

This object shall indicate the type of motor attached to and driven by the drive device. Table 7 specifies the value definition, Table 8 specifies the object description, and Table 9 specifies the entry description.

Table 7 – Value definition

Value	CANopen name	Other names
0000_h	Non-standard motor	-
0001_h	Phase modulated DC motor	-
0002_h	Frequency controlled DC motor	-
0003_h	PM synchronous motor	-
0004_h	FC synchronous motor	AC synchronous sinewave wound field
0005_h	Switched reluctance motor	AC synchronous reluctance switched
0006_h	Wound rotor induction motor	AC asynchronous induction polyphase wound rotor
0007_h	Squirrel cage induction motor	AC asynchronous induction squirrel cage
0008_h	Stepper motor	AC synchronous step
0009_h	Micro-step stepper motor	-
$000A_h$	Sinusoidal PM BL motor	AC synchronous sinusoidal PM
$000B_h$	Trapezoidal PM BL motor	AC synchronous brushless PM trapezoidal
$000C_h$	AC synchronous reluctance sync	-
$000D_h$	DC commutator PM	-
$000E_h$	DC commutator wound field series	-
$000F_h$	DC commutator wound field shunt	-
0010_h	DC commutator wound field compound	-
0011_h to $7FFE_h$	reserved	-
$7FFF_h$	No motor type assigned	-
8000_h to $FFFF_h$	Manufacturer-specific	-

Table 8 – Object description

Attribute	Value
Index	6402_h
Name	Motor type
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 9 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 7
Default value	Manufacturer-specific

6.3.2 Object 6403_h: Motor catalogue number

This object shall indicate the motor catalogue number (nameplate number) provided by the motor manufacturer. If the number is not assigned yet, this object shall indicate this by /0 (empty string). Table 10 specifies the object description, and Table 11 specifies the entry description.

Table 10 – Object description

Attribute	Value
Index	6403_h
Name	Motor catalogue number
Object code	Variable
Data type	Visible String
Category	Optional

Table 11 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Visible String
Default value	Manufacturer-specific

6.3.3 Object 6404_h: Motor manufacturer

This object shall indicate the name of the motor manufacturer. If the name is not assigned yet, this object shall indicate this by /0 (empty string). Table 12 specifies the object description, and Table 13 specifies the entry description.

Table 12 – Object description

Attribute	Value
Index	6404_h
Name	Motor manufacturer
Object code	Variable
Data type	Visible String
Category	Optional

Table 13 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Visible String
Default value	Manufacturer-specific

6.3.4 Object 6405_h: http motor catalogue address

This object shall indicate the assigned web-address of the motor catalogue. If the address is not assigned yet, this object shall indicate this by /0 (empty string). Table 14 specifies the object description, and Table 15 specifies the entry description.

Table 14 – Object description

Attribute	Value
Index	6405_h
Name	http motor catalogue address
Object code	Variable
Data type	Visible String
Category	Optional

Table 15 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Visible String
Default value	Manufacturer-specific

6.3.5 Object 6406_h: Motor calibration date

This object shall indicate the assigned date of the last motor inspection. If the date is not assigned yet, this object shall indicate this by a value of 0. Table 16 specifies the object description, and Table 17 specifies the entry description.

Table 16 – Object description

Attribute	Value
Index	6406_h
Name	Motor calibration date
Object code	Variable
Data type	Time of Day
Category	Optional

Table 17 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	0_d or Time of Day
Default value	Manufacturer-specific

6.3.6 Object 6407_h: Motor service period

This object shall indicate the assigned motor service period. If the period is not assigned yet, this object shall indicate this by $0000\ 0000_h$. The value shall be given in multiples of hours. Table 18 specifies the object description, and Table 19 specifies the entry description.

Table 18 – Object description

Attribute	Value
Index	6407_h
Name	Motor service period
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 19 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

6.3.7 Object 6503_h: Drive catalogue number

This object shall indicate the assigned manufacturer's drive catalogue number (nameplate number). Table 20 specifies the object description, and Table 21 specifies the entry description.

Table 20 – Object description

Attribute	Value
Index	6503_h
Name	Drive catalogue number
Object code	Variable
Data type	Visible String
Category	Optional

Table 21 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	No
Default value	/0 (empty string)

6.3.8 Object 6505_h: http drive catalogue address

This object shall indicate the assigned web address of the drive manufacturer. If the address is not assigned yet, this object shall indicate this by /0 (empty string). Table 22 specifies the object description, and Table 23 specifies the entry description.

Table 22 – Object description

Attribute	Value
Index	6505_h
Name	http drive catalogue address
Object code	Variable
Data type	Visible String
Category	Optional

Table 23 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	No
Default value	Manufacturer-specific

7 Error codes and error behaviour

7.1 Error codes

Emergency messages are triggered by internal errors and severe warnings detected within the drive device. They are defined in detail in the /CiA402-3/. They shall contain the 16-bit

error code. Error codes from $xx00_h$ to $xx7F_h$ are defined in /CiA301/ or in Table 24. Error codes between $xx80_h$ and $xxFF_h$ are used manufacturer-specific.

Table 24 – Error codes

Error code	Meaning
2110_h	Short circuit/earth leakage (input)
2120_h	Earth leakage (input)
2121_h	Earth leakage phase L1
2122_h	Earth leakage phase L2
2123_h	Earth leakage phase L3
2130_h	Short circuit (input)
2131_h	Short circuit phases L1-L2
2132_h	Short circuit phases L2-L3
2133_h	Short circuit phases L3-L1
2211_h	Internal current no.1
2212_h	Internal current no.2
2213_h	Over-current in ramp function
2214_h	Over-current in the sequence
2220_h	Continuous over current (device internal)
2221_h	Continuous over current no.1
2222_h	Continuous over current no.2
2230_h	Short circuit/earth leakage (device internal)
2240_h	Earth leakage (device internal)
2250_h	Short circuit (device internal)
2310_h	Continuous over current
2311_h	Continuous over current no.1
2312_h	Continuous over current no.2
2320_h	Short circuit/earth leakage (motor-side)
2330_h	Earth leakage (motor-side)
2331_h	Earth leakage phase U
2332_h	Earth leakage phase V
2333_h	Earth leakage phase W
2340_h	Short circuit (motor-side)
2341_h	Short circuit phases U-V
2342_h	Earth leakage phase V-W
2343_h	Earth leakage phase W-U
2350_h	Load level fault (I^2t, thermal state)
2351_h	Load level warning (I^2t, thermal state)
3110_h	Mains over-voltage
3111_h	Mains over-voltage phase L1
3112_h	Mains over-voltage phase L2

Error code	Meaning
3113_h	Mains over-voltage phase L3
3120_h	Mains under-voltage
3121_h	Mains under-voltage phase L1
3122_h	Mains under-voltage phase L2
3123_h	Mains under-voltage phase L3
3130_h	Phase failure
3131_h	Phase failure L1
3132_h	Phase failure L2
3133_h	Phase failure L3
3134_h	Phase sequence
3140_h	Mains frequency
3141_h	Mains frequency too great
3142_h	Mains frequency too small
3210_h	DC link over-voltage
3211_h	Over-voltage no. 1
3212_h	Over voltage no. 2
3220_h	DC link under-voltage
3221_h	Under-voltage no. 1
3222_h	Under-voltage no. 2
3230_h	Load error
3310_h	Output over-voltage
3311_h	Output over-voltage phase U
3312_h	Output over-voltage phase V
3313_h	Output over-voltage phase W
3320_h	Armature circuit
3321_h	Armature circuit interrupted
3330_h	Field circuit
3331_h	Field circuit interrupted
4110_h	Excess ambient temperature
4120_h	Too low ambient temperature
4130_h	Temperature supply air
4140_h	Temperature air outlet
4210_h	Excess temperature device
4220_h	Too low temperature device
4300_h	Temperature drive
4310_h	Excess temperature drive
4320_h	Too low temperature drive
4400_h	Temperature supply
4410_h	Excess temperature supply

Error code	Meaning
4420_h	Too low temperature supply
5100_h	Supply
5110_h	Supply low voltage
5111_h	U1 = supply ±15V
5112_h	U2 = supply +24 V
5113_h	U3 = supply +5 V
5114_h	U4 = manufacturer-specific
5115_h	U5 = manufacturer-specific
5116_h	U6 = manufacturer-specific
5117_h	U7 = manufacturer-specific
5118_h	U8 = manufacturer-specific
5119_h	U9 = manufacturer-specific
5120_h	Supply intermediate circuit
5200_h	Control
5210_h	Measurement circuit
5220_h	Computing circuit
5300_h	Operating unit
5400_h	Power section
5410_h	Output stages
5420_h	Chopper
5430_h	Input stages
5440_h	Contacts
5441_h	Contact 1 = manufacturer-specific
5442_h	Contact 2 = manufacturer-specific
5443_h	Contact 3 = manufacturer-specific
5444_h	Contact 4 = manufacturer-specific
5445_h	Contact 5 = manufacturer-specific
5450_h	Fuses
5451_h	S1 = I1
5452_h	S2 = I2
5453_h	S3 = I3
5454_h	S4 = manufacturer-specific
5455_h	S5 = manufacturer-specific
5456_h	S6 = manufacturer-specific
5457_h	S7 = manufacturer-specific
5458_h	S8 = manufacturer-specific
5459_h	S9 = manufacturer-specific
5500_h	Hardware memory
5510_h	RAM

Error code	Meaning
5520_h	ROM/EPROM
5530_h	EEPROM
6010_h	Software reset (watchdog)
6301_h to $630F_h$	Data record no. 1 to no. 15
6310_h	Loss of parameters
6320_h	Parameter error
7100_h	Power
7110_h	Brake chopper
7111_h	Failure brake chopper
7112_h	Over current brake chopper
7113_h	Protective circuit brake chopper
7120_h	Motor
7121_h	Motor blocked
7122_h	Motor error or commutation malfunc.
7123_h	Motor tilted
7200_h	Measurement circuit
7300_h	Sensor
7301_h	Tacho fault
7302_h	Tacho wrong polarity
7303_h	Resolver 1 fault
7304_h	Resolver 2 fault
7305_h	Incremental sensor 1 fault
7306_h	Incremental sensor 2 fault
7307_h	Incremental sensor 3 fault
7310_h	Speed
7320_h	Position
7400_h	Computation circuit
7500_h	Communication
7510_h	Serial interface no. 1
7520_h	Serial interface no. 2
7600_h	Data storage (external)
8300_h	Torque control
8311_h	Excess torque
8312_h	Difficult start up
8313_h	Standstill torque
8321_h	Insufficient torque
8331_h	Torque fault
8400_h	Velocity speed controller
8500_h	Position controller

Error code	Meaning
8600_h	Positioning controller
8611_h	Following error
8612_h	Reference limit
8700_h	Sync controller
8800_h	Winding controller
8900_h	Process data monitoring
$8A00_h$	Control
$F001_h$	Deceleration
$F002_h$	Sub-synchronous run
$F003_h$	Stroke operation
$F004_h$	Control
$FF00_h$ to $FFFF_h$	Manufacturer-specific

7.2 Error behavior

The communication system may support an object specifying to which network management state the drive device shall transit, when a communication error or a severe device-internal error is detected. When the PDS FSA transits into *Error* state, this shall be regarded as a severe device-internal failure.

8 Controlling the power drive system

8.1 General

The PDS FSA is an abstraction to define the behavior of a black box as a control device experiences the PDS. It defines the application behavior of the PDS. Due to the requirement that a PDS provides local control even when the communication network is not working properly, the communication FSA as defined in the communication system mapping specifications and the PDS FSA are only loosely coupled.

Figure 2 specifies how the PDS may be operated locally or via the network remotely. The PDS is operated by local signals (not in the scope of this specification) and by the controlword sent by the control device via the network. The state of the PDS is reported by the statusword produced by the drive device. The FSA is also controlled by error detection signals.

The PDS FSA defines the PDS status and the possible control sequence of the PDS. A single state represents a special internal or external behavior. The state of the PDS also determines which commands are accepted. For example, it is only possible to start a point-to-point move when the drive is in the *operation enabled* state.

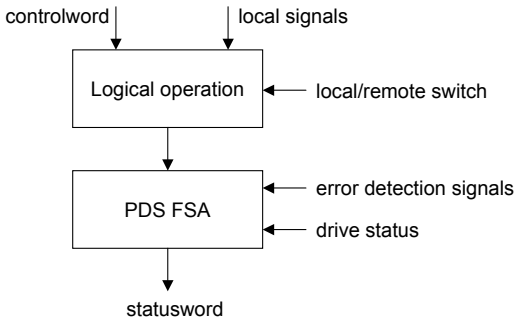


Figure 2 – Remote and local control

8.2 Finite state automaton

Figure 3 specifies the PDS FSA with respect to control of the power electronics as a result of user commands and internal drive faults.

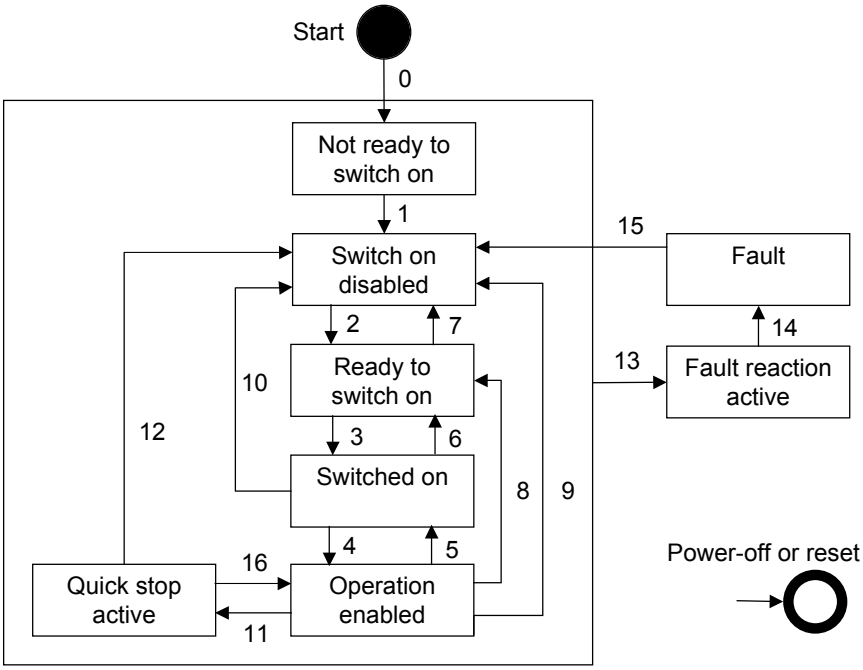


Figure 3 – Power drive system finite state automaton

The FSA states shall support the functions as shown in Table 25. The *start* state shall be a pseudo state indicating the start when the FSA is activated during the start-up sequence of the device drive's application software.

Table 25 – FSA states and supported functions

Function	FSA states							
	Not ready to switch on	Switch on disabled	Ready to switch on	Switched on	Operation enabled	Quick stop active	Fault reaction active	Fault
Brake applied, if present	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes
Low-level power applied	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Function	FSA states							
	Not ready to switch on	Switch on disabled	Ready to switch on	Switched on	Operation enabled	Quick stop active	Fault reaction active	Fault
High-level power applied	Yes/No	Yes/No	Yes/No	Yes	Yes	Yes	Yes	Yes/No
Drive function enabled	No	No	No	No	Yes	Yes	Yes	No
Configuration allowed	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

If in the *quick stop active* state the quick stop option code is set to 5, 6, 7 or 8, the drive device shall not leave this state, but it may transit to the *operation enabled* state with the *Enable operation* command.

The drive device shall support the transitions and actions as given in Table 26. The events shall initiate the transition. The transition shall be terminated, after the action has been performed.

Table 26 – Transition events and actions

Transition	Event(s)	Action(s)
0	Automatic transition after power-on or reset application	Drive device self-test and/or self initialisation shall be performed.
1	Automatic transition	Communication shall be activated.
2	Shutdown command from control device or local signal	None
3	Switch on command received from control device or local signal	The high-level power shall be switched on, if possible.
4	Enable operation command received from control device or local signal	The drive function shall be enabled and all internal set-points cleared.
5	Disable operation command received from control device or local signal	The drive function shall be disabled.
6	Shutdown command received from control device or local signal	The high-level power shall be switched off, if possible.
7	Quick stop or disable voltage command from control device or local signal	None
8	Shutdown command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
9	Disable voltage command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
10	Disable voltage or quick stop command from control device or local signal	The high-level power shall be switched off, if possible.
11	Quick stop command from control device or local signal	The quick stop function shall be started.
12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
13	Fault signal (see also /CiA402-3/)	The configured fault reaction function shall be executed.
14	Automatic transition	The drive function shall be disabled; the high-level power shall be switched off, if possible.

Transition	Event(s)	Action(s)
15	Fault reset command from control device or local signal	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the Fault state, the Fault reset bit in the controlword shall be cleared by the control device.
16	Enable operation command from control device, if the quick stop option code is 5, 6, 7, or 8	The drive function shall be enabled.
NOTE It is not recommended to support transition 16.		

If a state transition is requested, the related actions shall be processed completely before transitioning to the new state. Example: In *operation enabled* state, when the *disable operation* command is received, the drive device shall stay in the *operation enabled* state until the disable operation function (see object $605C_h$) is completed.

Drive devices able to control the contactor for the mains may switch the high-level power. If the high-level power is switched-off, the motor shall be free to rotate if not braked.

Drive function is disabled implies no energy shall be supplied to the motor. Target or set-point values (e.g. torque, velocity, position) shall be not processed.

Drive function is enabled implies that energy may be supplied to the motor. Target or set-point values shall be processed.

If a fault is detected in the drive device, there shall be a transition to the *fault reaction active* state. In this state, the PDS shall execute a special fault reaction. After the execution of this fault reaction, the drive device shall switch automatically to the *fault* state. This state shall only be left by the fault reset command, but only if the fault is not active any more.

In case of fatal error, the drive device is not longer able to control the motor, so that an immediate switch-off of the drive device is necessary.

The behaviour of drive disabling, quick stop, halt, and fault reaction functions is configurable by means of configuration objects defined in 8.4.

NOTE If a brake is present, the high-level power is switched off after a delay time in order to apply the brake.

8.3 Modes of operation

The PDS behaviour depends on the activated mode of operation. The PDS may implement several modes of operation. 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.

The control device writes to the *modes of operation* object in order to select the operation mode. The drive device provides the *modes of operation display* object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behaviour. The switching between the modes of operation implies no automatic reconfiguration of COBs for real-time data transmission.

Therefore, the PDS may limit mode switching in one or some PDS FSA state(s). Mode switching may also be limited to the 'local control' function; this means it is not possible to select the operation mode via the network.

The following modes of operation are described in this part of profile specification:

- Profile position mode
- Homing mode
- Interpolated position mode
- Profile velocity mode (e.g. servo drives)
- Torque profile mode
- Velocity mode (e.g. frequency converter)
- Cyclic sync position mode
- Cyclic sync velocity mode
- Cyclic sync torque mode

With the exception of the 'Homing mode', the listed modes of operation deal with set-points. In addition to this, manufacturer-specific modes of operation may also be implemented. These are not limited to set-points.

Figure 4 shows the general relations between target, reference, effort, and actual values.

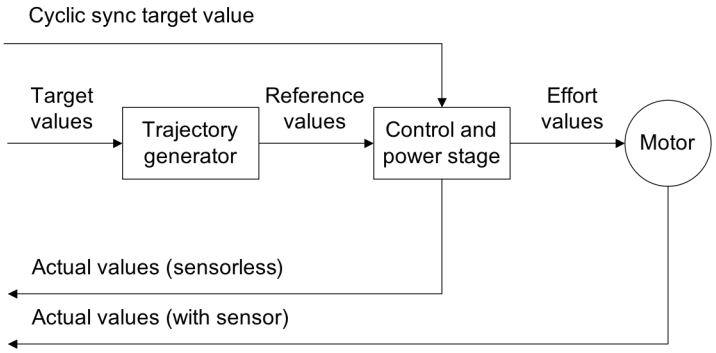


Figure 4 – Relation between different value parameters

8.4 Detailed object specifications

8.4.1 Object 6040_h: Controlword

This object shall indicate the received command controlling the PDS FSA. It shall be structured as defined in Figure 5. The bits 7, 3, 2, 1, and 0 shall be supported. The other bits may be supported. The commands shall be coded as given in Table 27.

15 – 11	10	9	8	7	6 – 4	3	2	1	0
ms	*r*	*oms*	*h*	*fr*	*oms*	*eo*	*qs*	*ev*	*so*
MSB									LSB

LEGEND: ms = manufacturer-specific; r = reserved; oms = operation mode specific; h = halt; fr = fault reset; eo = enable operation; qs = quick stop; ev = enable voltage; so = switch on

Figure 5 – Value definition

Table 27 – Command coding

Command	Bits of the *controlword*					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3 + 4 (NOTE)
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
NOTE Automatic transition to Enable operation state after executing SWITCHED ON state functionality.						

Bits 9, 6, 5, and 4 of the controlword are operation mode specific. The halt function (bit 8) behaviour is operation mode specific. If the bit is 1, the commanded motion shall be interrupted, the PDS shall behave as defined in the halt option code. After releasing the halt function, the commanded motion shall be continued if possible.

The bit 10 is reserved for further use; it shall be set to 0. The bits 11, 12, 13, 14, and 15 are manufacturer-specific.

Table 28 specifies the object description, and Table 29 specifies the entry description.

Table 28 – Object description

Attribute	Value
Index	6040_h
Name	Controlword
Object code	Variable
Data type	Unsigned16
Category	Mandatory

Table 29 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 27
Default value	Device and operation mode specific

8.4.2 Object 6041_h: Statusword

This object shall provide the status of the PDS FSA. The object shall be structured as defined in Figure 6. The bits 10, 9, and 6 to 0 shall be supported. The other bits may be supported. The bit combinations defined in Table 30 shall code the PDS FSA states.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms		*oms*		*ila*	*tr*	*rm*	*ms*	*w*	*sod*	*qs*	*ve*	*f*	*oe*	*so*	*rtso*

MSB LSB

LEGEND: ms = manufacturer-specific; oms = operation mode specific; ila = internal limit active; tr = target reached; rm = remote; w = warning; sod = switch on disabled; qs = quick stop; ve = voltage enabled; f = fault; oe = operation enabled; so = switched on; rtso = ready to switch on

Figure 6 – Value definition

Table 30 – State coding

Statusword	PDS FSA state
xxxx xxxx x0xx 0000_b	Not ready to switch on
xxxx xxxx x1xx 0000_b	Switch on disabled
xxxx xxxx x01x 0001_b	Ready to switch on
xxxx xxxx x01x 0011_b	Switched on
xxxx xxxx x01x 0111_b	Operation enabled
xxxx xxxx x00x 0111_b	Quick stop active
xxxx xxxx x0xx 1111_b	Fault reaction active
xxxx xxxx x0xx 1000_b	Fault

If bit 4 (voltage enabled) of the statusword is 1, this shall indicate that high voltage is applied to the PDS.

If bit 5 (quick stop) of the statusword is 0, this shall indicate that the PDS is reacting on a quick stop request.

If bit 7 (warning) of the statusword is 1, this shall indicate the presence of a warning condition. Warning is not an error or fault (examples: temperature limit exceeded, job refused). The status of the PDS FSA shall not be changed. The cause of the warning may be given in the *fault code parameter* object ($603F_h$).

If bit 9 (remote) of the statusword is 1, this shall indicate that the controlword is processed. If it is 0 (local), this shall indicate that the controlword is not processed. Nevertheless, the PDS may provide actual values, and the PDS may accept COB for configuration data transmission for other parameter objects.

If bit 10 (target reached) of the statusword is 1, this shall indicate that the PDS has reached the set-point. The set-point is operation mode specific and is defined in detail in the corresponding clauses of this part of profile specification. Bit 10 shall also be set to 1, if the operation mode has been changed. The change of a target value by software shall alter this bit. If quick stop option code is 5, 6, 7 or 8, bit 10 shall be set to 1, when the quick stop operation is finished and the PDS is halted. If halt occurred and the PDS has halted then bit 10 shall be set to 1, too.

If bit 11 (internal limit active) of the statusword is 1, this shall indicate that an internal limit is active (example: position range limit). The internal limits are manufacturer-specific.

Bit 13 and bit 12 of the statusword are operation mode specific.

Bit 14 and bit 15 are manufacturer-specific.

Table 31 specifies the object description, and Table 32 specifies the entry description.

Table 31 – Object description

Attribute	Value
Index	6041_h
Name	Statusword
Object code	Variable
Data type	Unsigned16
Category	Mandatory

Table 32 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	See Table 30
Default value	No

8.4.3 Object $603F_h$: Error code

This object shall provide the error code of the last error, which occurred in the drive device. Table 24 specifies the value definition, Table 33 specifies the object description, and Table 34 specifies the entry description.

NOTE In CANopen networks, this object provides the same information as the lower 16-bit of sub-index 01_h of the pre-defined error field (1003_h).

Table 33 – Object description

Attribute	Value
Index	$603F_h$
Name	Error code
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 34 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	See Table 24
Default value	No

8.4.4 Object 6007_h: Abort connection option code

This object shall indicate what action shall be performed when one of the following events occurres: bus-off, heartbeat, life guarding, NMT stopped state entered, reset application, and reset communication. Table 35 specifies the value definition, Table 36 specifies the object description, and Table 37 specifies the entry description.

Table 35 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	No action
+1	Fault signal
+2	Disable voltage command
+3	Quick stop command
+4 to +32 767	reserved

Table 36 – Object description

Attribute	Value
Index	6007_h
Name	Abort connection option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 37 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 35
Default value	+1

8.4.5 Object $605A_h$: Quick stop option code

This object shall indicate what action is performed when the quick stop function is executed. The slow down ramp is the deceleration value of the used mode of operations. Table 38

specifies the value definition, Table 39 specifies the object description, and Table 40 specifies the entry description.

Table 38 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Disable drive function
+1	Slow down on slow down ramp and transit into Switch On Disabled
+2	Slow down on quick stop ramp and transit into Switch On Disabled
+3	Slow down on current limit and transit into Switch On Disabled
+4	Slow down on voltage limit and transit into Switch On Disabled
+5	Slow down on slow down ramp and stay in Quick Stop Active
+6	Slow down on quick stop ramp and stay in Quick Stop Active
+7	Slow down on current limit and stay in Quick Stop Active
+8	Slow down on voltage limit and stay in Quick Stop Active
+9 to +32 767	reserved

Table 39 – Object description

Attribute	Value
Index	$605A_h$
Name	Quick stop option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 40 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 38
Default value	+2

8.4.6 Object $605B_h$: Shutdown option code

This object shall indicate what action is performed if there is a transition from Operation Enabled state to Ready To Switch On state. The slow down ramp is the deceleration value of the used mode of operations. Table 41 specifies the value definition, Table 42 specifies the object description, and Table 43 specifies the entry description.

Table 41 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Disable drive function (switch-off the drive power stage)
+1	Slow down with slow down ramp; disable of the drive function
+2 to +32 767	reserved

Table 42 – Object description

Attribute	Value
Index	$605B_h$
Name	Shutdown option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 43 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 41
Default value	0

8.4.7 Object $605C_h$: Disable operation option code

This object shall indicate what action is performed if there is a transition from Operation Enabled state to Switched on state. The slow down ramp is the deceleration value of the used mode of operations. Table 44 specifies the value definition, Table 45 specifies the object description, and Table 46 specifies the entry description.

Table 44 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Disable drive function (switch-off the drive power stage)
+1	Slow down with slow down ramp; disable of the drive function
+2 to +32 767	reserved

Table 45 – Object description

Attribute	Value
Index	$605C_h$
Name	Disable operation option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 46 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 44
Default value	+1

8.4.8 Object $605D_h$: Halt option code

This object shall indicate what action is performed when the halt function is executed. The slow down ramp is the deceleration value of the used mode of operations. Table 47 specifies the value definition, Table 48 specifies the object description, and Table 49 specifies the entry description.

Table 47 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Reserved
+1	Slow down on slow down ramp and stay in Operation Enabled
+2	Slow down on quick stop ramp and stay in Operation Enabled
+3	Slow down on current limit and stay in Operation Enabled
+4	Slow down on voltage limit and stay in Operation Enabled
+5 to +32 767	Reserved

Table 48 – Object description

Attribute	Value
Index	$605D_h$
Name	Halt option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 49 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 47
Default value	+1

8.4.9 Object $605E_h$: Fault reaction option code

This object shall indicate what action is performed when fault is detected in the PDS. The slow down ramp is the deceleration value of the used mode of operations. Table 50 specifies the value definition, Table 51 specifies the object description, and Table 52 specifies the entry description.

Table 50 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Disable drive function, motor is free to rotate
+1	Slow down on slow down ramp
+2	Slow down on quick stop ramp
+3	Slow down on current limit
+4	Slow down on voltage limit
+5 to +32 767	reserved

Table 51 – Object description

Attribute	Value
Index	$605E_h$
Name	Fault reaction option code
Object code	Variable
Data type	Integer16
Category	Optional

Table 52 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 50
Default value	+2

8.4.10 Object 6060_h: Modes of operation

This object shall indicate the requested operation mode. Table 53 specifies the value definition, Table 54 specifies the object description, and Table 55 specifies the entry description.

NOTE This object shows only the value of the requested operation mode, the actual operation mode of the PDS is reflected in the object modes of operation display.

Table 53 – Value definition

Value	Definition
-128 to -1	Manufacturer-specific operation modes
0	No mode change/no mode assigned
+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	Cyclic sync position mode
+9	Cyclic sync velocity mode
+10	Cyclic sync torque mode
+11 to +127	reserved

Table 54 – Object description

Attribute	Value
Index	6060_h
Name	Modes of operation
Object code	Variable
Data type	Integer8
Category	Optional

Table 55 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 53
Default value	0

8.4.11 Object 6061_h: Modes of operation display

This object shall provide the actual operation mode. Table 53 specifies the value definition, Table 56 specifies the object description, and Table 57 specifies the entry description.

Table 56 – Object description

Attribute	Value
Index	6061_h
Name	Modes of operation display
Object code	Variable
Data type	Integer8
Category	Optional

Table 57 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	See Table 53
Default value	No

8.4.12 Object 6502_h: Supported drive modes

This object shall provide information on the supported drive modes. Figure 7 specifies the value definition, Table 58 specifies the object description, and Table 59 specifies the entry description.

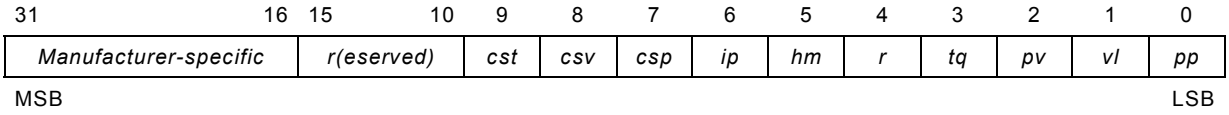


Figure 7 – Value definition

cst, csv, csp, ip, hm, tq, pv, vl, and pp bits:

1 = mode is supported 0 = mode is not supported

manufacturer-specific bits:

1 = function is supported 0 = function is not supported

r(eserved) bits: 0

Table 58 – Object description

Attribute	Value
Index	6502_h
Name	Supported drive modes
Object code	Variable
Data type	Unsigned32
Category	Mandatory

Table 59 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	See Figure 7
Default value	No

9 Factor group

9.1 General

In some drive device applications several sensor resolution values and ratio values are needed. They may make use for the objects defined in this clause.

The relation between the user-defined units and the internal units is calculated by the following equation:

$$position\ actual\ value = \frac{position\ internal\ value\ \times feed\ constant}{position\ encoder\ resolution\ \times gear\ ratio}$$

9.2 Detailed object definitions

9.2.1 Object $608F_h$: Position encoder resolution

This object shall indicate the configured encoder increments and number of motor revolutions. The position encoder resolution shall be calculated by the following formula:

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

All values shall be dimensionless. Table 60 specifies the object description, and Table 61 specifies the entry description.

Table 60 – Object description

Attribute	Value
Index	$608F_h$
Name	Position encoder resolution
Object code	Array
Data type	Unsigned32
Category	Optional

Table 61 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	01_h
Description	Encoder increments
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	02_h
Description	Motor revolutions
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)

9.2.2 Object 6090_h: Velocity encoder resolution

This object shall indicate the configured encoder increments per second and motor revolutions per second. The velocity encoder resolution shall be calculated by the following formula:

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

All values shall be dimensionless. Table 62 specifies the object description, and Table 63 specifies the entry description.

Table 62 – Object description

Attribute	Value
Index	6090_h
Name	Velocity encoder resolution
Object code	Array
Data type	Unsigned32
Category	Optional

Table 63 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	01_h
Description	Encoder increments per second
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	02_h
Description	Motor revolutions per second
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)

9.2.3 Object 6091_h: Gear ratio

This object shall indicate the configured number of motor shaft revolutions and number of driving shaft revolutions. The gear ratio shall be calculated by the following formula:

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

All values shall be dimensionless. Table 64 specifies the object description, and Table 65 specifies the entry description.

Table 64 – Object description

Attribute	Value
Index	6091_h
Name	Gear ratio
Object code	Array
Data type	Unsigned32
Category	Optional

Table 65 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	01_h
Description	Motor revolutions
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	02_h
Description	Shaft revolutions
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)

9.2.4 Object 6092_h: Feed constant

This object shall indicate the configured feed constant, this is the measurement distance per one revolution of the output shaft of the gearbox. The feed constant shall be calculated by the following formula:

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

The feed shall be given in user-defined position units, and the driving shaft revolution shall be dimensionless. Table 66 specifies the object description, and Table 67 specifies the entry description.

Table 66 – Object description

Attribute	Value
Index	6092_h
Name	Feed constant
Object code	Array
Data type	Unsigned32
Category	Optional

Table 67 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	01_h
Description	Feed
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)
Sub-Index	02_h
Description	Shaft revolutions
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific (but not equal to 0)

9.2.5 Object $607E_h$: Polarity

This object shall indicate if the position demand value shall be multiplied by 1 of by –1. The polarity flag shall have no influence on the homing mode. The position polarity bit shall be used only for profile position (*pp*) mode and cyclic sync position mode (*csp*). The velocity polarity bit shall be used only for profile velocity (*pv*) mode and cyclic sync velocity mode

(*csv*). Figure 8 specifies the value definition, Table 68 specifies the object description, and Table 69 specifies the entry description.

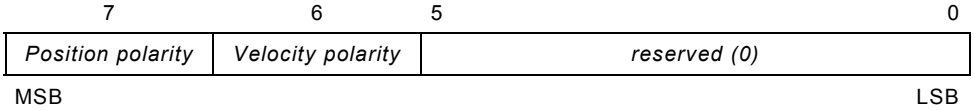


Figure 8 – Value definition

The polarity bits shall be coded as follows: 0_b = multiply by 1 and 1_b = multiply by -1

Table 68 – Object description

Attribute	Value
Index	$607E_h$
Name	Polarity
Object code	Variable
Data type	Unsigned8
Category	Optional

Table 69 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Figure 8
Default value	00_h

10 Profile position mode

10.1 General information

The overall structure for this mode is shown in Figure 9. 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 (see 12.3.1). These two function blocks are optionally controlled by individual parameter sets.

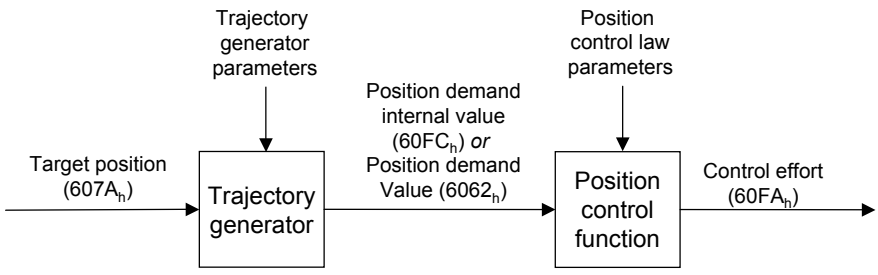


Figure 9 – Trajectory generator and position control function

At the input to the trajectory generator, parameters may have optional limits applied before being normalised to internal units. The simplest form of a trajectory generator is just to pass

through a target position and to transform it to a position demand internal value with internal units (increments) only. Figure 10 defines the detailed structure of the trajectory generator.

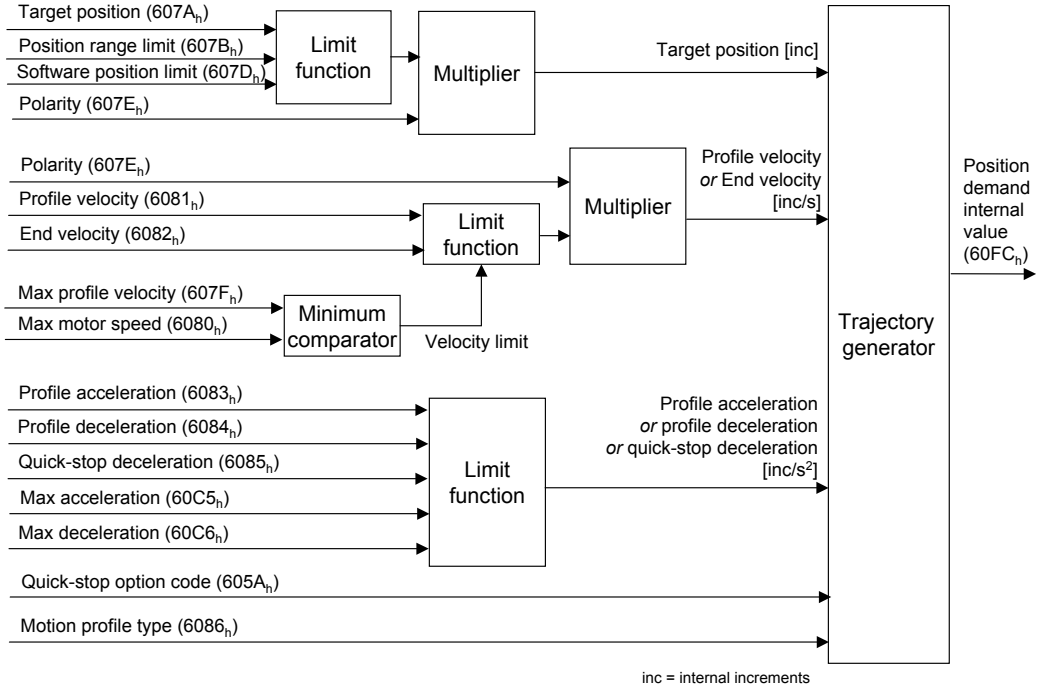


Figure 10 – Trajectory generator for profile position mode

10.2 Functional description

10.2.1 General

The setting of set-points is controlled by the timing of the *new set-point* bit and the *change set immediately* bit in the controlword as well as the *set-point acknowledge* bit in the statusword.

If the *change set immediately* bit of the controlword is set to 1, a single set-point is expected by the drive device. If the *change set immediately* bit of the controlword is set to 0, a set of set-points is expected by the drive device.

After a set-point is applied to the drive device, the control device signals that the set-point is valid by a rising edge of the *new set-point* bit in the controlword. The drive device sets the set-point acknowledge bit in the statusword to 1, and afterwards, the drive device signals with the *set-point acknowledge* bit set to 0 its ability to accept new set-points. An example is shown in Figure 11.

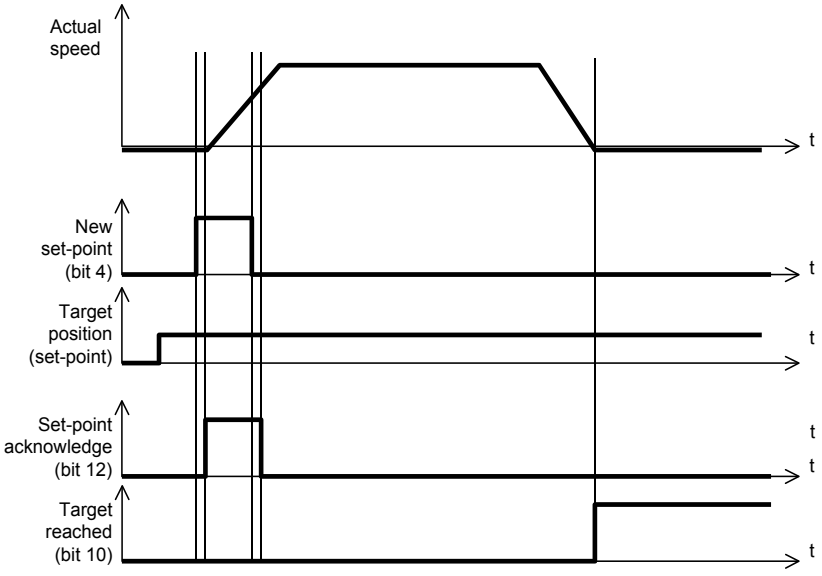


Figure 11 – Set-point example

If one set-point is still in progress and a new one is validated, two methods of handling are supported: *single set-point* (*change set immediately* bit of controlword is 1) and *set of set-points* (*change set immediately* bit of controlword is 0).

10.2.2 Single set-point

When a set-point is in progress and a new set-point is validated by the new set-point (bit 4) in the controlword, the new set-point shall be processed immediately. The handshaking procedure shown in Figure 12 is used for the single set-point method.

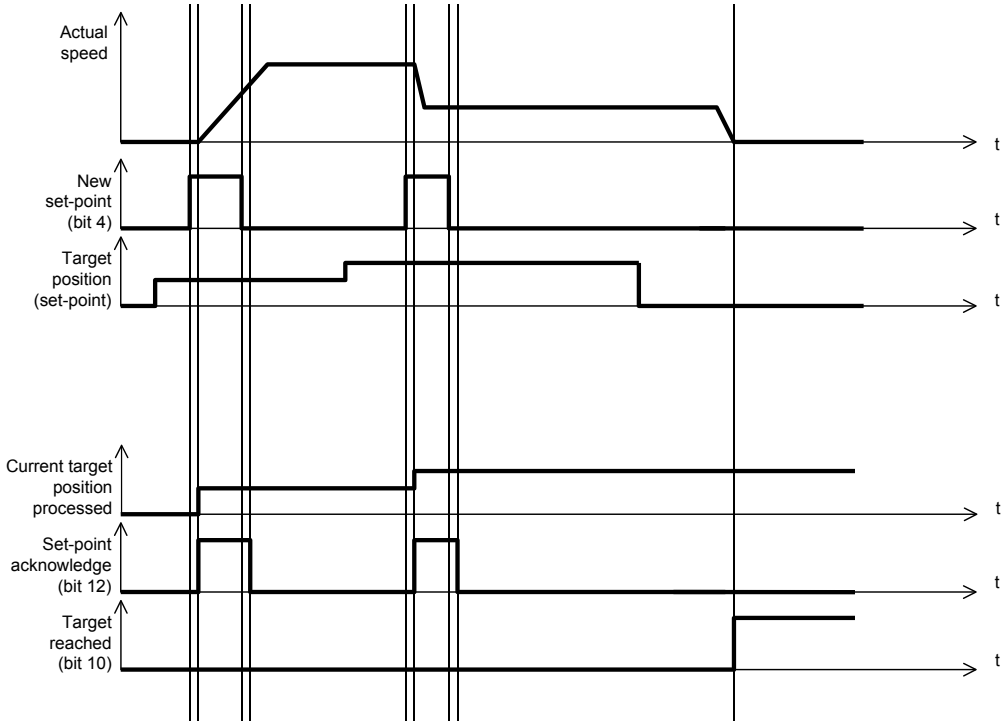


Figure 12 – Handshaking procedure for the single set-point method

10.2.3 Set of set-points

When a set-point is in progress and a new set-point is validated by the new set-point (bit 4) in the controlword, the new set-point shall be processed only after the previous has been reached. The handshaking procedure shown in Figure 13 is used for the set of set-points method. The additional grey line segment in the graph 'actual speed' shows the actual speed if the *change of set point* bit (bit 9) is set to 1.

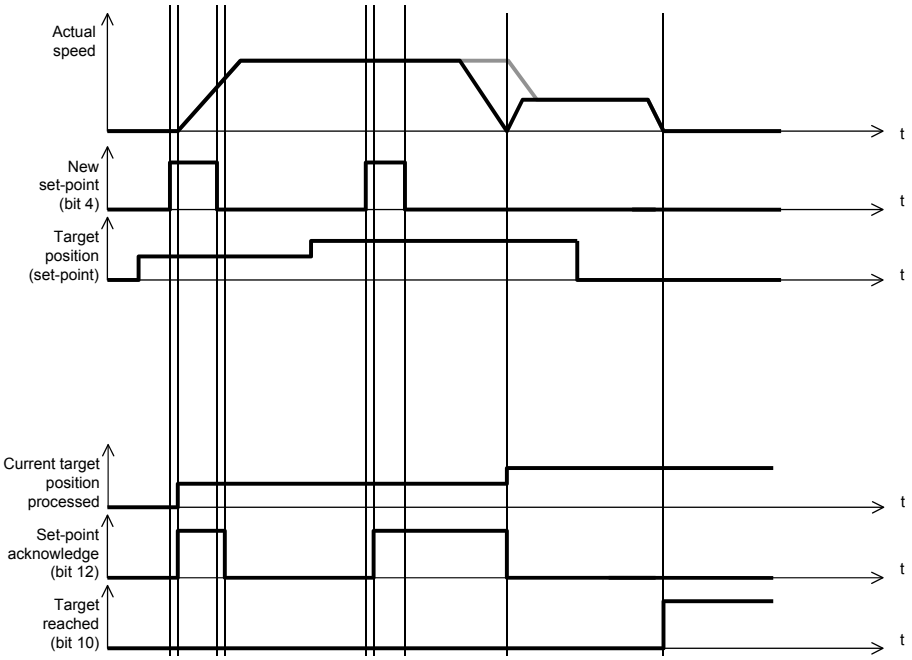


Figure 13 – Handshaking procedure for the set of set-points method

If a drive device supports set of set-points, a minimum of two set-points are available, a set-point that is currently been processed an d a buffered set-point. The set-points are handled as shown in Figure 14.

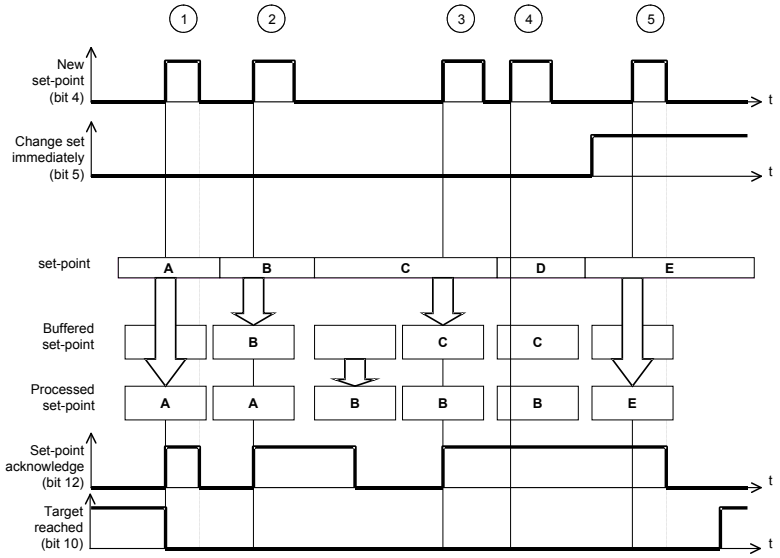


Figure 14 – Set-point handling for two set-points

New set-points are buffered in the set-point list as long as free set-points are available in the drive device. If no set-point is in progress, the new set-point shall become active immediately

(1). If a set-point is in progress, the new set-point shall be stored in the first set-point buffer that is free (2 + 3).

If all set-point buffers are busy (*set-point acknowledge´*bit is 1), the reaction depends on the *change set immediately* bit. If the *change set immediately* bit is set to 1, the new set-point shall be processed immediately as single set-point. All previously loaded set-points shall be discarded (5).

The *target reached* bit shall remain 0 until all set-points are processed.

10.3 General definitions

The internal software limits shall not be exceeded by external settings configured by the user.

10.4 Use of controlword and statusword

The profile position mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 15 shows the structure of the controlword. If no positioning is in progress, the rising edge of bit 4 shall start the positioning of the axis. In case a positioning is in progress, the definitions given in Table 70 shall be used. Table 71 defines the values for bit 6 and 8 of the controlword.

NOTE It is assumed that the target position is edge-triggered 0->1 otherwise the drive could set immediately new values, which leads to unexpected behaviour.

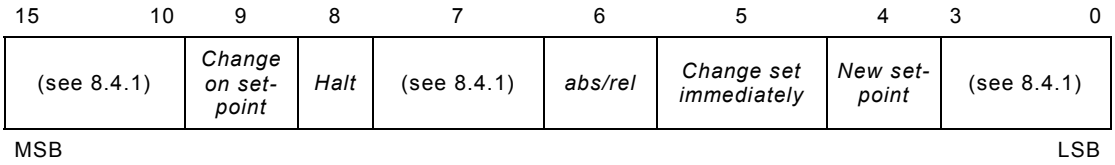


Figure 15 – Controlword for profile position (pp) mode

Table 70 – Definition of bit 4, bit 5, and bit 9

Bit 9	Bit 5	Bit 4	Definition
0	0	0 -> 1	Positioning shall be completed (target reached) before the next one gets started (see Figure 11 and Figure 13)
X	1	0 -> 1	Next positioning shall be started immediately (see Figure 11 and Figure 12)
1	0	0 -> 1	Positioning with the current profile velocity up to the current setpoint shall be proceeded and then next positioning (see Figure 11 and Figure 13) shall be applied

Table 71 – Definition of bit 6 and bit 8

Bit	Value	Definition
6	0	*Target position* shall be an absolute value
	1	*Target position* shall be a relative value (depending on object $60F2_h$)
8	0	Positioning shall be executed or continued
	1	Axis shall be stopped accordingly to halt option code ($605D_h$)

Figure 16 shows the structure of the statusword. Table 72 defines the values for bit 10, bit 12, and bit 13.

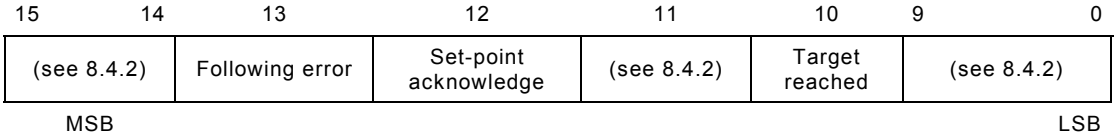


Figure 16 – Statusword for profile position (pp) mode

Table 72 – Definition of bit 10, bit 12, and bit 13

Bit	Value	Definition
10	0	Halt (Bit 8 in controlword) = 0: *Target position* not reached Halt (Bit 8 in controlword) = 1: Axis decelerates
	1	Halt (Bit 8 in controlword) = 0: *Target position* reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0
12	0	Previous setpoint already processed, waiting for new setpoint
	1	Previous setpoint still in process, setpoint overwriting shall be accepted
13	0	No following error
	1	Following error

10.5 Detailed object definitions

10.5.1 Object $607A_h$: Target position

This object shall indicate the commanded 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 value of this object shall be interpreted as absolute or relative depending on the ‘abs/rel' flag in the controlword. It shall be given in user-defined position units and shall be converted to position increments. Table 73 specifies the object description, and Table 74 specifies the entry description.

Table 73 – Object description

Attribute	Value
Index	$607A_h$
Name	Target position
Object code	Variable
Data type	Integer32
Category	Optional; mandatory if *pp, pc* or *csp* is supported

Table 74 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

10.5.2 Object $607B_h$: Position range limit

This object shall indicate the configured maximal and minimal position range limits. It shall limit the numerical range of the input value. On reaching or exceeding these limits, the input value shall wrap automatically to the other end of the range. Wrap-around of the input value may be prevented by setting software position limits as defined in software position limit object ($607D_h$). The values shall be given in user-defined position units. Table 75 specifies the object description, and Table 76 specifies the entry description.

Table 75 – Object description

Attribute	Value
Index	$607B_h$
Name	Position range limit
Object code	Array
Data type	Integer32
Category	Optional

Table 76 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific
Sub-Index	01_h
Description	Min position range limit
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	Max position range limit
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

10.5.3 Object $607D_h$: Software position limit

This object shall indicate the configured maximal and minimal software position limits. These parameters shall define the absolute position limits for the position demand value and the position actual value. Every new target position shall be checked against these limits. The limit positions shall be always relative to the machine home position. Before being compared with the target position, they shall be corrected internally by the home offset as follows:

corrected min position limit = min position limit - home offset

corrected max position limit = max position limit - home offset

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

The limit positions shall be given in user-defined position units (same as target position). Table 77 specifies the object description, and Table 78 specifies the entry description.

Table 77 – Object description

Attribute	Value
Index	$607D_h$
Name	Software position limit
Object code	Array
Data type	Integer32
Category	Optional

Table 78 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	Manufacturer-specific
Sub-Index	01_h
Description	Min position limit
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

Attribute	Value
Sub-Index	02_h
Description	Max position limit
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

10.5.4 Object $607F_h$: Max profile velocity

This object shall indicate the configured maximal allowed velocity in either direction during a profiled motion. The value shall be given in the very same physical unit as the *profile velocity* object (6081_h). Table 79 specifies the object description, and Table 80 specifies the entry description.

Table 79 – Object description

Attribute	Value
Index	$607F_h$
Name	Max profile velocity
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 80 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.5 Object 6080_h: Max motor speed

This object shall indicate the configured maximal allowed speed for the motor in either direction. It is used to protect the motor and is taken from the motor data sheet. The value shall be given in rotations per minute (rpm). Table 81 specifies the object description, and Table 82 specifies the entry description.

Table 81 – Object description

Attribute	Value
Index	6080_h
Name	Max motor speed
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 82 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.6 Object 6081_h: Profile velocity

This object shall indicate the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and shall be valid for both directions of motion. allowed velocity in either direction during a profiled motion. The value shall be given in user-defined speed units. ~~It shall be converted to position increments per second using the *velocity encoder factor* object.~~ Table 83 specifies the object description, and Table 84 specifies the entry description.

Table 83 – Object description

Attribute	Value
Index	6081_h
Name	Profile velocity
Object code	Variable
Data type	Unsigned32
Category	Conditional: mandatory if *pp* is supported

Table 84 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.7 Object 6082_h: End velocity

This object shall indicate the configured velocity, which the drive shall have on reaching the target position. Normally, the drive stops at the target position, i.e. the end velocity = 0. The value shall be given in the same physical unit as the *profile velocity* object (6081_h). Table 85 specifies the object description, and Table 86 specifies the entry description.

Table 85 – Object description

Attribute	Value
Index	6082_h
Name	End velocity
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 86 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	$0000\ 0000_h$

10.5.8 Object 6083_h: Profile acceleration

This object shall indicate the configured acceleration. The value shall be given in user-defined acceleration units; ~~it shall be converted to position increments per square second (s^2) using the normalising factors (see Clause 9~~). Table 87 specifies the object description, and Table 88 specifies the entry description.

Table 87 – Object description

Attribute	Value
Index	6083_h
Name	Profile acceleration
Object code	Variable
Data type	Unsigned32
Category	Conditional: mandatory if *pp* or *pv* is supported

Table 88 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.9 Object 6084_h: Profile deceleration

This object shall indicate the configured deceleration. If this parameter is not supported, then the *profile acceleration* object (6083_h) value shall be used for deceleration, too. The value shall be given in the same physical units as *profile acceleration* object (6083_h). Table 89 specifies the object description, and Table 90 specifies the entry description.

Table 89 – Object description

Attribute	Value
Index	6084_h
Name	Profile deceleration
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 90 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.10 Object 6085_h: Quick stop deceleration

This object shall indicate the configured deceleration used to stop the motor when the quick stop function is activated and the *quick stop code* object ($605A_h$) is set to 2 or 6. The quick stop deceleration is also used if the *fault reaction code* object ($605E_h$) is 2 and the *halt option code* object ($605D_h$) is 2. The value shall be given in the same physical unit as *profile acceleration* object (6083_h). Table 91 specifies the object description, and Table 92 specifies the entry description.

Table 91 – Object description

Attribute	Value
Index	6085_h
Name	Quick stop deceleration
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 92 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.11 Object 6086_h: Motion profile type

This object shall indicate the configured type of motion profile used to perform a profiled motion. Table 93 specifies the value definition, Table 94 specifies the object description, and Table 95 specifies the entry description.

Table 93 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Linear ramp (trapeziodal profile)
+1	Sin^2 ramp
+2	Jerk-free ramp
+3	Jerk-limited ramp
+4 to +32 767	Reserved

Table 94 – Object description

Attribute	Value
Index	6086_h
Name	Motion profile type
Object code	Variable
Data type	Integer16
Category	Optional

Table 95 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	0

10.5.12 Object $60A3_h$: Profile jerk use

This object shall indicate the configured number of sub-indices used in the *profile jerk* object ($60A4_h$) for the jerk profile movement. If this object is not implemented, the *profile jerk* object shall be used as it is implemented. The value shall be dimensionless, the value of FF_h shall

indicate that the profile jerk use is not configured. Table 96 specifies the object description, and Table 97 specifies the entry description.

Table 96 – Object description

Attribute	Value
Index	$60A3_h$
Name	Profile jerk use
Object code	Variable
Data type	Unsigned8
Category	Optional

Table 97 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	01_h to 06_h and FF_h
Default value	Manufacturer-specific

10.5.13 Object $60A4_h$: Profile jerk

This object shall indicate the configured set of jerk parameters that shall be used during the profile movement. Figure 17 shows the defined jerks (A, B, C, D, E, and F). The values shall be given in user-defined jerk units. Table 98 specifies the value assignment to jerks depending of the value of *profile jerk use* object ($60A3_h$). If object $60A3_h$ is not implemented, the sub-index 00_h shall be used to assign the values given in the other sub-indices to the jerks. Table 99 specifies the object description, and Table 100 specifies the entry description.

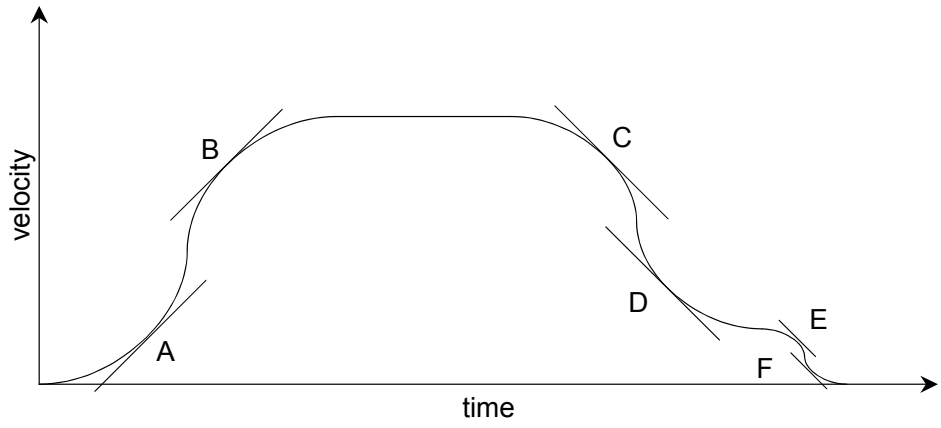


Figure 17 – Velocity/time diagram with jerk positions

Table 98 – Value assignments

Value in $60A3_h$ or sub-index 00_h of $60A4_h$ if $60A3_h$ is not implemented	Value assignment to jerks					
	A	B	C	D	E	F
01_h	01_h	01_h	01_h	01_h	-	-
02_h	01_h	01_h	02_h	02_h	-	-
04_h	01_h	03_h	02_h	04_h	-	-
06_h	01_h	03_h	02_h	04_h	05_h	06_h

Table 99 – Object description

Attribute	Value
Index	$60A4_h$
Name	Profile jerk
Object code	Array
Data type	Unsigned32
Category	Optional

Table 100 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	01_h, 02_h, 04_h, or 06_h,
Default value	Manufacturer-specific
Sub-Index	01_h
Description	Profile jerk 1
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

Attribute	Value
Sub-Index	02_h
Description	Profile jerk 2
Entry Category	Optional
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific
to	
Sub-Index	06_h
Description	Profile jerk 6
Entry Category	Optional
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.14 Object $60C5_h$: Max acceleration

This object shall indicate the configured maximal acceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. The value shall be given in user-defined acceleration physical units. Table 101 specifies the object description, and Table 102 specifies the entry description.

Table 101 – Object description

Attribute	Value
Index	$60C5_h$
Name	Max acceleration
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 102 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

10.5.15 Object $60C6_h$: Max deceleration

This object shall indicate the configured maximal deceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. The value shall be given in the same physical unit as the *max acceleration* object ($60C5_h$). Table 103 specifies the object description, and Table 104 specifies the entry description.

Table 103 – Object description

Attribute	Value
Index	$60C6_h$
Name	Max deceleration
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 104 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

11 Homing mode

11.1 General information

This clause 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.

11.2 Functional description

Figure 18 shows the defined input objects as well as the output objects. The user may 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 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.

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.

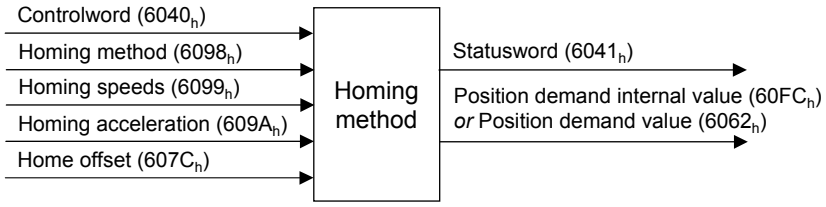


Figure 18 – Homing mode function

By choosing a homing method, the following behaviour is determined: 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.

An encircled number in the figures Figure 19 to Figure 26 indicates the code for selection of this homing position. The direction of movement is also indicated.

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 case, that a limit switch has reached the drive shall move in the other direction to leave the position.

In the diagrams of homing sequences shown below, the encoder count increases as the axis'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 drive device has several methods to choose from, using the homing method.

11.3 General definitions

11.3.1 Method 1: Homing on negative limit switch and index pulse

Using this method as shown in Figure 19, the initial direction of movement shall be leftward if the negative limit switch is inactive (here: low). The home position shall be at the first index pulse to the right of the position where the negative limit switch becomes inactive.

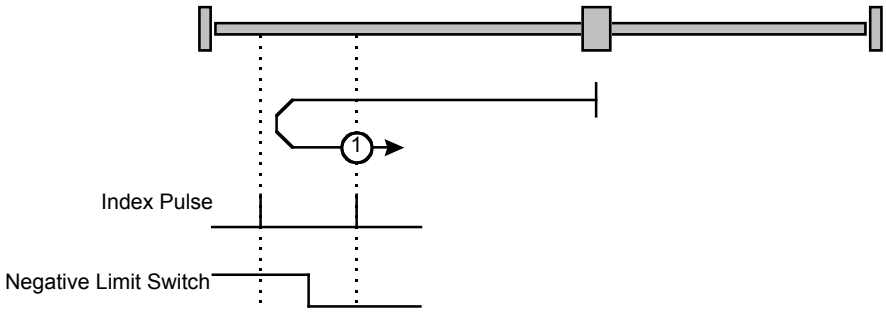


Figure 19 – Homing on negative limit switch and index pulse

11.3.2 Method 2: Homing on positive limit switch and index pulse

Using this method as shown in Figure 20, the initial direction of movement shall be rightward if the positive limit switch is inactive (here: low). The position of home shall be at the first index pulse to the left of the position where the positive limit switch becomes inactive.

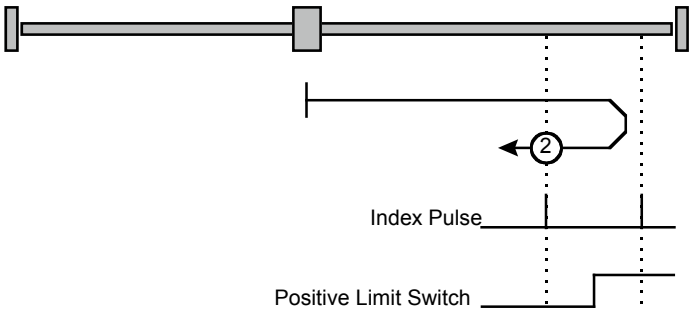


Figure 20 – Homing on positive limit switch and index pulse

11.3.3 Method 3 and 4: Homing on positive home switch and index pulse

Using these methods as shown in Figure 21, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be 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 situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

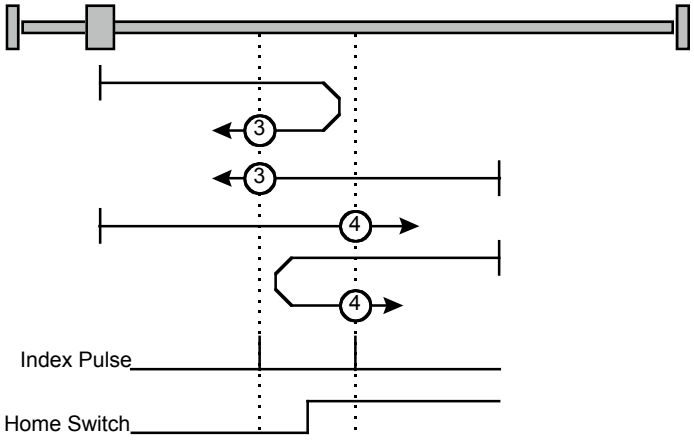


Figure 21 – Homing on positive home switch and index pulse

11.3.4 Method 5 and 6: Homing on negative home switch and index pulse

Using these methods as shown in Figure 22, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be 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 situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

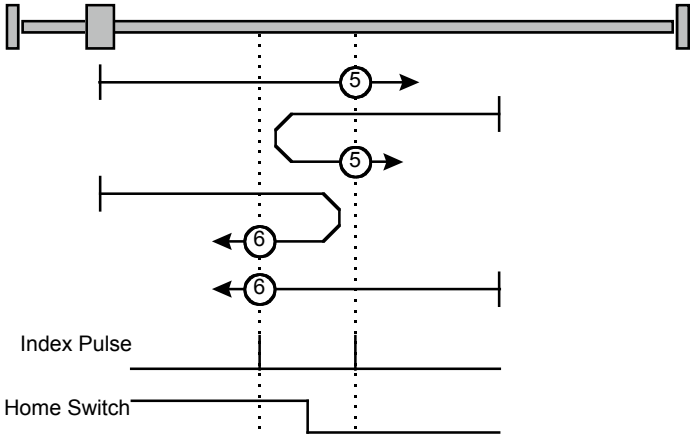


Figure 22 – Homing on negative home switch and index pulse

11.3.5 Method 7 to 14: Homing on home switch and index pulse

These methods use a home switch, which is active over only a portion of the travel, in effect the switch has a 'momentary' action as the axis's position sweeps past the switch. Using the methods 7 to 10, the initial direction of movement shall be to the right, and using methods 11 to 14, the initial direction of movement shall be to the left except if the home switch is active at the start of the motion. In this case, the initial direction of motion shall be dependent on the edge being sought. The home position shall be at the index pulse on either side of the rising or falling edges of the home switch, as shown in Figure 23 and Figure 24. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

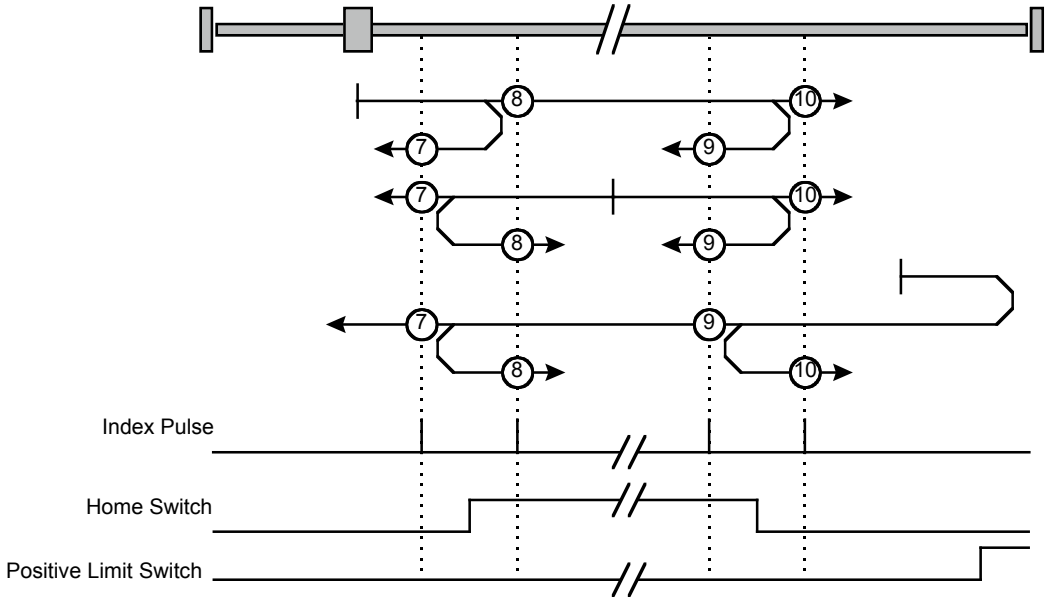


Figure 23 – Homing on home switch and index pulse – positive initial motion

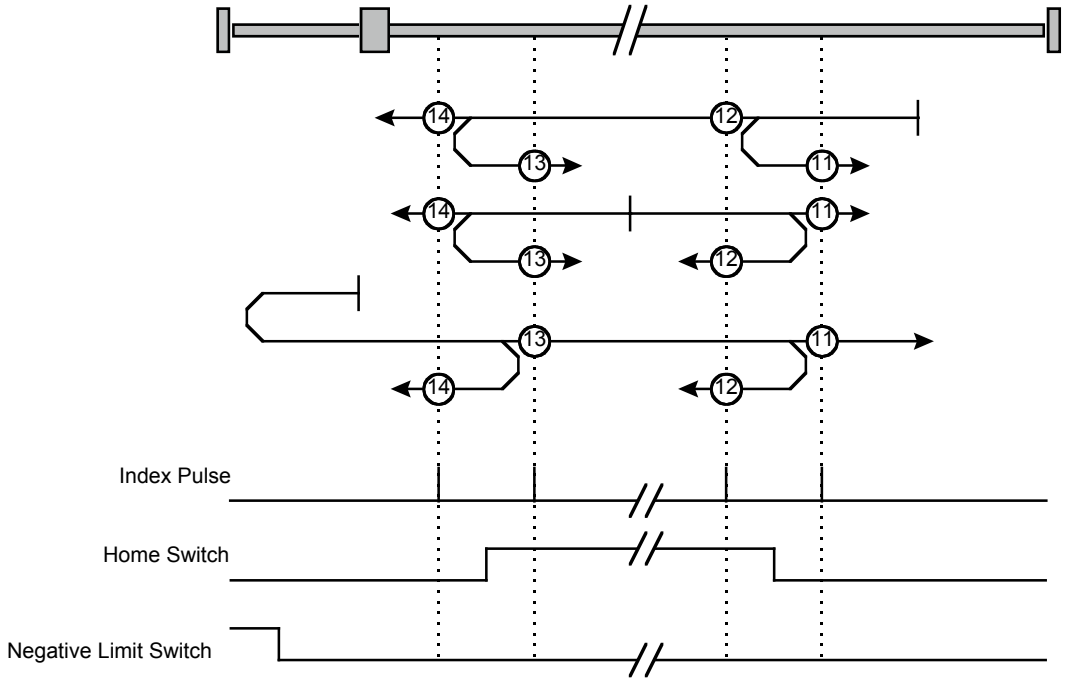


Figure 24 – Homing on home switch and index pulse – negative initial motion

11.3.6 Method 15 and 16: Reserved

These methods are reserved.

11.3.7 Method 17 to 30: Homing without 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 Figure 25.

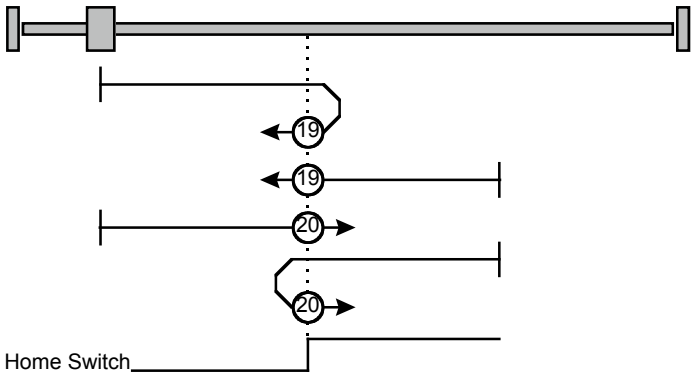


Figure 25 – Homing on positive home switch

11.3.8 Method 31 and 32: Reserved

These methods are reserved.

11.3.9 Method 33 and 34: Homing on index pulse

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction as shown in Figure 26.

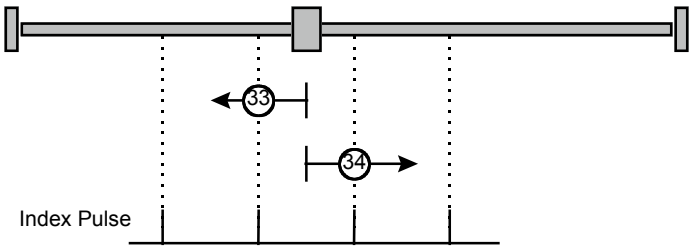


Figure 26 – Homing on index pulse

11.3.10 Method 35: Homing on index pulse

In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in *operational enabled* state.

11.3.11 Method 36: Homing with touch-probe

In this method, the position is not sampled by the control device, but by the drive device itself. When the switch is triggered, the corresponding actual position together with the switch signal shall be reported.

11.4 Use of controlword and statusword

The homing mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 27 shows the structure of the controlword. Table 105 defines the values for bit 4 and 8 of the controlword.

15 … 9	8	7	6	5	4	3 … 0
(see 8.4.1)	*Halt*	(see 8.4.1)	reserved (0)		*Homing operation start*	(see 8.4.1)
MSB						LSB

Figure 27 – Controlword for homing mode

Table 105 – Definition of bit 4 and bit 8

Bit	Value	Definition
4	0	Do not start homing procedure
	1	Start or continue homing procedure
8	0	Enable bit 4
	1	Stop axis according to halt option code ($605D_h$)

Figure 28 shows the structure of the statusword. Table 106 defines the values for bit 10, bit 12, and bit 13.

15 … 14	13	12	11	10	9 … 0
(see 8.4.2)	Homing error	Homing attained	(see 8.4.2)	Target reached	(see 8.4.2)
MSB					LSB

Figure 28 – Statusword for homing mode

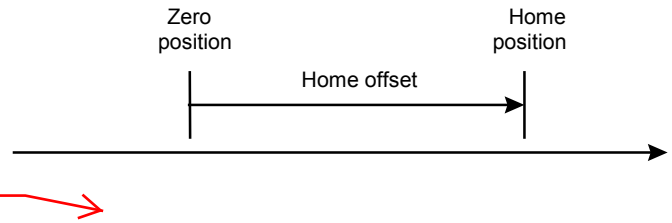
Table 106 – Definition of bit 10, bit 12, and bit 13

Bit 13	Bit 12	Bit 10	Definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not started
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing procedure is completed successfully
1	0	0	Homing error occurred, velocity is not 0
1	0	1	Homing error occurred, velocity is 0
1	1	X	reserved

11.5 Detailed object definitions

11.5.1 Object $607C_h$: Home offset

This object shall indicate the configured difference between the zero position for the application and the machine home position (found during homing). 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 Figure 29. If this object is not implemented, then the home offset shall be regarded as zero. The value of this object shall be given in user-defined position units. Negative values shall indicate the opposite direction.



NOTE The activation of a new value of the object home offset is manufacturer-specific. It is recommended to apply the new value only while the drive is in homing mode.

Figure 29 – Home offset definition

Table 107 specifies the object description, and Table 108 specifies the entry description.

Table 107 – Object description

Attribute	Value
Index	$607C_h$
Name	Home offset
Object code	Variable
Data type	Integer32
Category	Optional

Table 108 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	0_d

11.5.2 Object 6098_h: Homing method

This object shall indicate the configured homing method that shall be used. Table 109 specifies the value definition, Table 110 specifies the object description, and Table 111 specifies the entry description.

Table 109 – Value definition

Value	Definition
-128_d to -1_d	Manufacturer-specific
0_d	No homing method assigned
$+1_d$	Method 1 shall be used
to	
$+35_d$	Method 35 shall be used
$+36_d$	Method 36 shall be used
$+37_d$ to $+127_d$	reserved

Table 110 – Object description

Attribute	Value
Index	6098_h
Name	Homing method
Object code	Variable
Data type	Integer8
Category	Conditional: mandatory if *hm* is supported

Table 111 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 109
Default value	Manufacturer-specific

11.5.3 Object 6099_h: Homing speeds

This object shall indicate the configured speeds used during homing procedure. The values shall be given in user-defined velocity units. Table 112 specifies the object description, and Table 113 specifies the entry description.

Table 112 – Object description

Attribute	Value
Index	6099_h
Name	Homing speeds
Object code	Array
Data type	Unsigned32
Category	Conditional: mandatory if *hm* is supported

Table 113 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	Speed during search for switch
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	Speed during search for zero
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

11.5.4 Object $609A_h$: Homing acceleration

This object shall indicate the configured acceleration and deceleration to be used during homing operation. The value shall be given in user-defined acceleration units. Table 114 specifies the object description, and Table 115 specifies the entry description.

Table 114 – Object description

Attribute	Value
Index	$609A_h$
Name	Homing acceleration
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 115 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

11.5.5 Object $60B8_h$: Touch probe function

This object shall indicate the configured function of the touch probe. Table 116 specifies the value definition, Table 117 specifies the object description, and Table 118 specifies the entry description.

Table 116 – Value definition

Bit	Value	Definition
0	0	Switch off touch probe 1
	1	Enable touch probe 1
1	0	Trigger first event
	1	continous
2	0	Trigger with touch probe 1 input
	1	Trigger with zero impulse signal or position encoder
3	0	Reserved
4	0	Switch off sampling at positive edge of touch probe 1
	1	Enable sampling at positive edge of touch probe 1
5	0	Switch off sampling at negative edge of touch probe 1
	1	Enable sampling at negative edge of touch probe 1
6, 7	-	User-defined (e.g. for testing)
8	0	Switch off touch probe 2
	1	Enable touch probe 2
9	0	Trigger first event
	1	Continous
10	0	Trigger with touch probe 2 input
	1	Trigger with zero impulse signal or position encoder
11	0	Reserved

Bit	Value	Definition
12	0	Switch off sampling at positive edge of touch probe 2
	1	Enable sampling at positive edge of touch probe 2
13	0	Switch off sampling at negative edge of touch probe 2
	1	Enable sampling at negative edge of touch probe 2
14, 15	-	User-defined (e.g. for testing)

Table 117 – Object description

Attribute	Value
Index	$60B8_h$
Name	Touch probe function
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 118 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 116
Default value	Manufacturer-specific

11.5.6 Object $60B9_h$: Touch probe status

This object shall provide the status of the touch probe. Table 119 specifies the value, Table 120 specifies the object description, and Table 121 specifies the entry description.

Table 119 – Value definition

Bit	Value	Definition
0	0	Touch probe 1 is switched off
	1	Touch probe 1 is enabled
1	0	Touch probe 1 no positive edge value stored
	1	Touch probe 1 positive edge position stored
2	0	Touch probe 1 no negative edge value stored
	1	Touch probe 1 negative edge position stored
3 to 5	0	Reserved
6, 7	-	User-defined (e.g. for testing)
8	0	Touch probe 2 is Switched off
	1	Touch probe 2 is Enabled
9	0	Touch probe 2 no positive edge value stored
	1	Touch probe 2 positive edge position stored
10	0	Touch probe 2 no negative edge value stored
	1	Touch probe 2 negative edge position stored
11 to 13	0	Reserved
14, 15	-	User-defined (e.g. for testing)

NOTE Bit 1 and bit 2 are set to 0_b when touch probe 1 is switched off (object $60B8_h$ bit 0 is 0_b). Bit 9 and 10 are set to 0_b when touch probe 2 is switched off (object $60B8_h$ bit 8 is 0_b).

Table 120 – Object description

Attribute	Value
Index	$60B9_h$
Name	Touch probe status
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 121 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	See Table 119
Default value	No

11.5.7 Object $60BA_h$: Touch probe pos1 pos value

This object shall provide the position value of the touch probe 1 at positive edge. The value shall be given in user-defined position units. Table 122 specifies the object description, and Table 123 specifies the entry description.

Table 122 – Object description

Attribute	Value
Index	$60BA_h$
Name	Touch probe pos1 pos value
Object code	Variable
Data type	Integer32
Category	Optional

Table 123 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

11.5.8 Object $60BB_h$: Touch probe pos1 neg value

This object shall provide the position value of the touch probe 1 at negative edge. The value shall be given in user-defined position units. Table 124 specifies the object description, and Table 125 specifies the entry description.

Table 124 – Object description

Attribute	Value
Index	$60BB_h$
Name	Touch probe pos1 neg value
Object code	Variable
Data type	Integer32
Category	Optional

Table 125 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

11.5.9 Object $60BC_h$: ~~Touch probe pos2 pos value~~ Touch probe 2 positive edge

This object shall provide the position value of the touch probe 2 at positive edge. The value shall be given in user-defined position units. Table 126 specifies the object description, and Table 127 specifies the entry description.

Table 126 – Object description

Attribute	Value
Index	$60BC_h$
Name	~~Touch probe pos2 pos value~~ Touch probe 2 positive edge
Object code	Variable
Data type	Integer32
Category	Optional

Table 127 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

11.5.10 Object $60BD_h$: ~~Touch probe pos2 neg value~~ Touch probe 2 negative edge

This object shall provide the position value of the touch probe 2 at negative edge. The value shall be given in user-defined position units. Table 128 specifies the object description, and Table 129 specifies the entry description.

Table 128 – Object description

Attribute	Value
Index	$60BD_h$
Name	~~Touch probe pos2 neg value~~ Touch probe 2 negative edge
Object code	Variable
Data type	Integer32
Category	Optional

Table 129 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12 Position control function

12.1 General information

For closed-loop position, the *position demand value* (as one of the outputs of the trajectory generator) and the output of the position detection unit (*position actual value*) like a resolver

or encoder are used input parameters. The behaviour of the closed-loop control is influenced by control parameters, which are externally applicable. To keep the loop stable, a relative limitation of the output using the previous *control effort* is optional. In order not to exceed the physical limits of a drive, an absolute limit function may be implemented for the control effort.

12.2 Functional description

Figure 30 shows the inputs and outputs of the position control function. 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 in the drive device. Especially in cascaded control structures, where a position control is followed by a torque control, for example the *control effort* of the position control loop is used as an input for a further calculation.

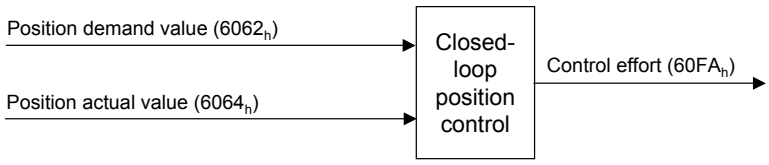


Figure 30 – Position control function

All values are transformed – if necessary – from user-defined units to normalised units such as increments.

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 shall result in setting bit 13 (*following error*) in the statusword to 1. This is shown in detail in Figure 31. Depending on the supported modes of operation (*pp*, *hm*, or *ip*) and on the capabilities of different categories of drives, only some of the mentioned input parameters may be necessary.

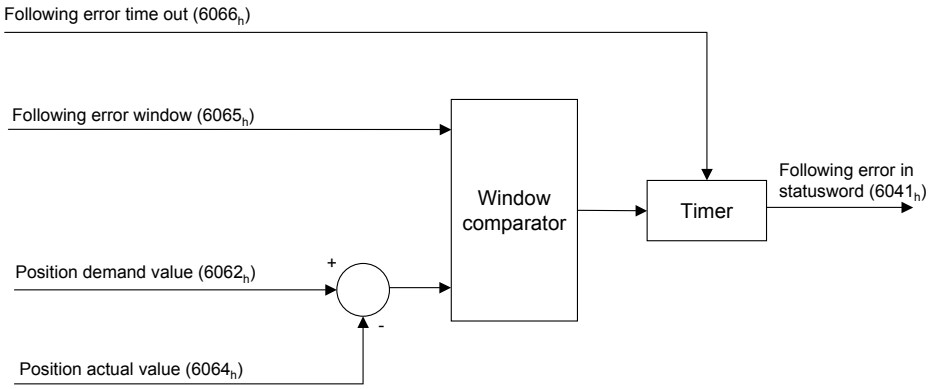


Figure 31 – Following error (functional overview)

The *position reached* function as shown in Figure 32 offers the possibility to define a position range around a position demand value to be regarded as valid. If a drive's position is within this area for a specified time – the position window time – the related control bit 10 target reached in the statusword shall be set to 1.

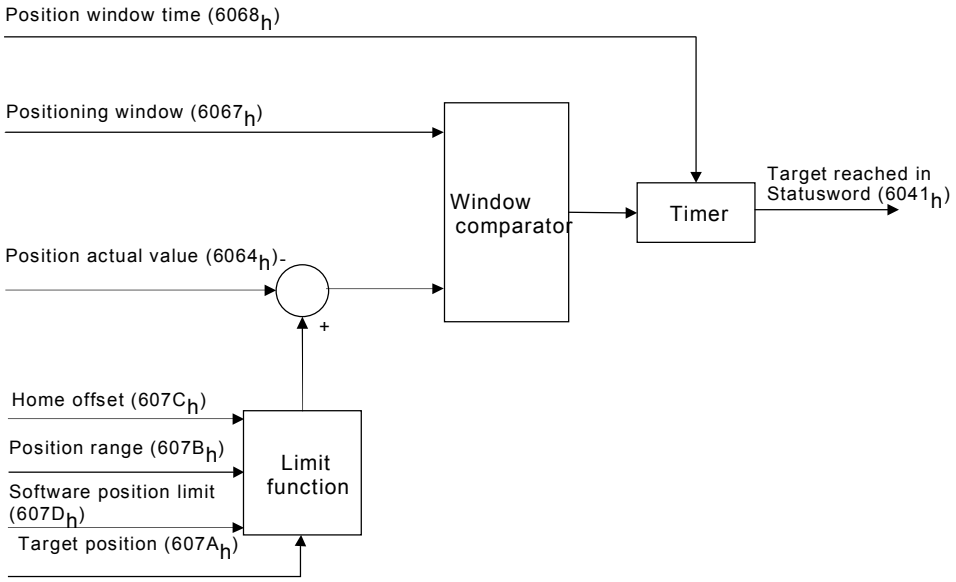


Figure 32 – Position reached (functional overview)

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

Figure 33 shows the definitions of the sub-function position reached. A window is defined for the accepted position range symmetrically around the target position. 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 shall be set to 1.

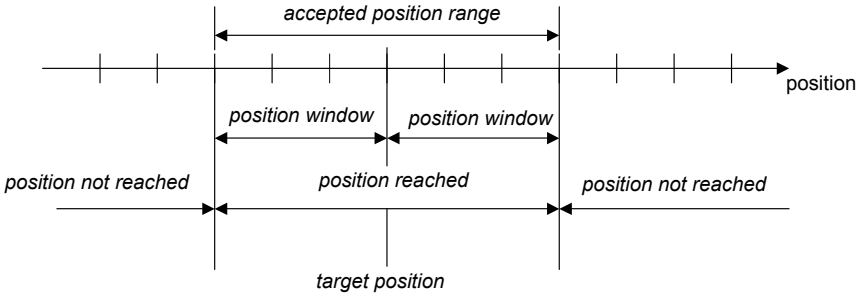


Figure 33 – Position reached (definitions)

Figure 34 shows the definitions of the sub-function following error in the profile position mode. A window is defined for the accepted following error tolerance symmetrically around the reference position. If a drive is situated out of the accepted position range for more than following error time out time, the bit following error (bit 13) in the statusword shall be set to 1.

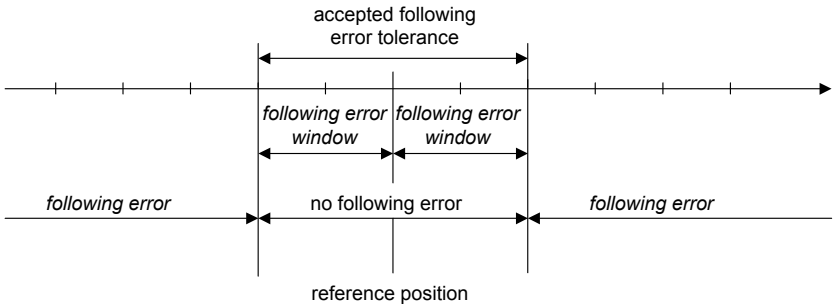


Figure 34 – Following error (definitions)

12.3 Detailed object definitions

12.3.1 Object 6062_h: Position demand value

This object shall provide the demanded position value. The value shall be given in user-defined position units. Table 130 specifies the object description, and Table 131 specifies the entry description.

Table 130 – Object description

Attribute	Value
Index	6062_h
Name	Position demand value
Object code	Variable
Data type	Integer32
Category	Optional

Table 131 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.2 Object 6063_h: Position actual internal value

This object shall provide the actual value of the position measurement device, which shall be one of the two input values of the closed-loop position control. If necessary, the data unit may be transformed from user-defined units to increments. The value shall be given in internal units. Table 132 specifies the object description, and Table 133 specifies the entry description.

Table 132 – Object description

Attribute	Value
Index	6063_h
Name	Position actual internal value
Object code	Variable
Data type	Integer32
Category	Optional

Table 133 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.3 Object 6064_h: Position actual value

This object shall provide the actual value of the position measurement device. The value shall be given in user-defined position units. Table 134 specifies the object description, and Table 135 specifies the entry description.

Table 134 – Object description

Attribute	Value
Index	6064_h
Name	Position actual value
Object code	Variable
Data type	Integer32
Category	~~Mandatory if *csp* is supported~~

Mandatory if *pp*, *ip*, *csp* is supported

Table 135 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.4 Object 6065_h: Following error window

This object shall indicate the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in user-

defined position units. If the value of the following error window is FFFF FFFF$_h$, the following control shall be switched off. Table 136 specifies the object description, and Table 137 specifies the entry description.

Table 136 – Object description

Attribute	Value
Index	6065$_h$
Name	Following errror window
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 137 – Entry description

Attribute	Value
Sub-Index	00$_h$
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

12.3.5 Object 6066$_h$: Following error time out

This object shall indicate the configured time for a following error condition, after that the bit 13 of the statusword shall be set to 1. The reaction of the drive when a following error occurs is manufacturer-specific. The value shall be given in ms. Table 138 specifies the object description, and Table 139 specifies the entry description.

Table 138 – Object description

Attribute	Value
Index	6066$_h$
Name	Following errror time out
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 139 – Entry description

Attribute	Value
Sub-Index	00$_h$
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

12.3.6 Object 6067_h: Position window

This object shall indicate the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position shall be regarded as having been reached. As the user mostly prefers to specify the position window in his application in user-defined units, the value is transformed into increments. The target position shall be handled in the same manner as in the *trajectory generator* concerning limiting functions and transformation into internal machine units before it may be used with this function. The value shall be given in user-defined position units. If the value of the position window is FFFF $FFFF_h$, the position window control shall be switched off.

Table 140 specifies the object description, and Table 141 specifies the entry description.

Table 140 – Object description

Attribute	Value
Index	6067_h
Name	Position window
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 141 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

12.3.7 Object 6068_h: Position window time

This object shall indicate the configured time, during which the actual position within the position window is measured. The value shall be given in ms. Table 142 specifies the object description, and Table 143 specifies the entry description.

Table 142 – Object description

Attribute	Value
Index	6068_h
Name	Position window time
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 143 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

12.3.8 Object $60F4_h$: Following error actual value

This object shall provide the actual value of the following error. The value shall be given in user-defined position units. Table 144 specifies the object description, and Table 145 specifies the entry description.

Table 144 – Object description

Attribute	Value
Index	$60F4_h$
Name	Following errror actual value
Object code	Variable
Data type	Integer32
Category	Optional

Table 145 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.9 Object $60FA_h$: Control effort

This object shall provide the control effort as the output of the position control loop. It is particular to the *position control function* that the notation of the control effort is mode-dependent and therefore not specified. ~~The value shall be given in user-defefined velocity units~~. Table 146 specifies the object description, and Table 147 specifies the entry description.

Replace: The value shall be given in user-defefined position units.

Table 146 – Object description

Attribute	Value
Index	$60FA_h$
Name	Control effort
Object code	Variable
Data type	Integer32
Category	Optional

Table 147 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.10 Object $60FC_h$: Position demand internal value

This object shall provide the output of the trajectory generator in profile position mode. This value shall be given in increments of the position encoder. Table 148 specifies the object description, and Table 149 specifies the entry description.

Table 148 – Object description

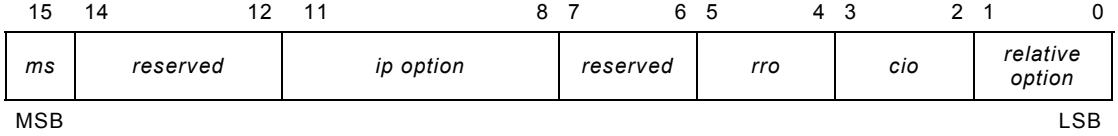
Attribute	Value
Index	$60FC_h$
Name	Position demand internal value
Object code	Variable
Data type	Integer32
Category	Optional

Table 149 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

12.3.11 Object $60F2_h$: Positioning option code

This object shall indicate the configured positioning behaviour as described by the *profile positioning mode* or the *interpolated positioning mode*. Figure 35 shows the defined object structure.



LEGEND ms = manufacturer-specific rro = request-response option cio = change immediately option

Figure 35 – Object structure

The *relative option* bits shall control the behaviour of positioning tasks in detail when the *abs_rel* bit (bit 6) of the controlword is set to 1 in *pp* mode. Table 150 shows the bit value definitions.

Table 150 – Value definition for bit 0 and bit 1

Bit 1	Bit 0	Definition
0	0	Positioning moves shall be performed relative to the preceding (internal absolute) target position (rsp. relative to 0 if there is no preceding target position) as described in 10.2
0	1	Positioning moves shall be performed relative to the actual position demand value (object $60FC_h$) – output of the trajectory generator
1	0	Positioning moves shall be performed relative to the position actual value (object 6064_h)
1	1	Reserved

The *change immediately option* bits shall control the behaviour of positioning tasks in detail when the *change_set_immediately* bit (bit 5) of the controlword is set to 1 in *pp* mode. Table 151 shows the bit value definitions.

Table 151 – Value definition for bit 2 and bit 3

Bit 3	Bit 2	Definition
0	0	The drive device shall readapt the actual motion to the new target position (considering potentially changed profile velocity and accelerations etc.) immediately as described in 10.2
0	1	The actually performed positioning task shall be continued (without attempting to stop on target position) and blended to the newly commanded task (considering potentially changed profile velocity and accelerations etc.) when target position is touched
1	0	Reserved
1	1	Reserved

The *request-response option* bits shall allow the drive device to release the *new_setpoint* bit (bit 4) of the controlword internally in order to avoid the need of setting this bit to 0 by the control device in *pp* mode. After internally releasing the *new_setpoint* bit, the drive device shall indicate the action to the control device by setting the *setpoint_acknowledgement* bit (bit 12) in the statusword to 0. Table 152 shows the bit value definitions.

Table 152 – Value definition for bit 4 and bit 5

Bit 5	Bit 4	Definition
0	0	The handshake as described in 10.2 shall be performed
0	1	The drive device shall release autonomously the *new setpoint* bit as soon as target is reached
1	0	The drive shall release autonomously the *new setpoint* bit as soon as able to accept new set-point data
1	1	Reserved

The *ip option* bits are reserved for defining the interpolated position mode. When the *manufacturer-specific* bit is set to 0, the function shall be not enabled; if it is set to 1, the manufacturer-specific function shall be enabled. The other reserved bits shall be set to 0.

Table 153 specifies the object description, and Table 154 specifies the entry description.

Table 153 – Object description

Attribute	Value
Index	$60F2_h$
Name	Positioning option code
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 154 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 150, Table 151, Table 152
Default value	0000_h

13 Interpolated position mode

13.1 General information

The interpolated position mode is used to control multiple coordinated axes or a single axis with the need for time-interpolation of set-point data. The interpolated position mode normally uses time synchronisation mechanisms 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 are manufacturer-specific.

For synchronous operation, the interpolation cycle time is defined by the object interpolation time period. Time synchronisation may be done by network dependent mechanisms. Each syncronisation cycle actuates the next data record if a valid data record is available.

For asynchronous operation, the interpolation time (for each time slice), may be included in the interpolation data record. If this is so, then the units for the interpolation time are still specified by the interpolation time index as for synchronous operation. The next data record shall be actuated as soon as the interpolation time expires and a valid data record is available.

The interpolated position mode allows the control deviced 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 maximum size of the input buffer may be read by the control device using the interpolation data configuration. The actual buffer size may be both written and read by the control device 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. Drive devices without input buffer capabilities shall accept at least one interpolation data item.

The interpolation data buffer may be implemented as a FIFO or a ring. The definition of a valid data record for each type of buffer shall be as follows:

- For the FIFO implementation, a valid data record is one that has not been actuated yet.

- For the Ring implementation, all data records within the actual buffer size are treated as valid data records, so interpolation data will continue to be actuated while ip enable is true.

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 shall 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.

The placement of the scaling and limiting of the interpolation data record in Figure 36 is for indication only. These functions may be performed during the input of the interpolation data record.

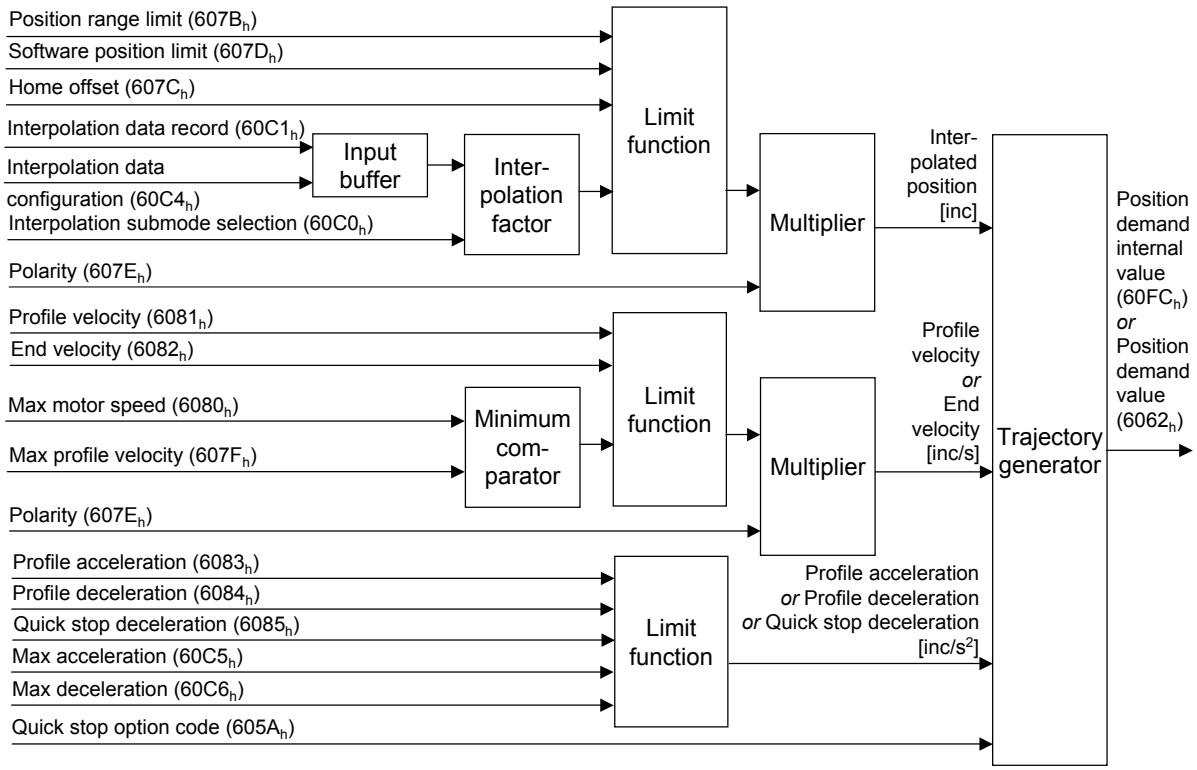


Figure 36 – Interpolation controller

13.2 Functional description

13.2.1 General

The manufacturer specifies the way the drive device handles the next valid interpolation data record. This may be in a way corresponding to the standard position mode, or might be a more complex algorithm. The standard method is to apply the new data immediately, after the next synchronisation signal in synchronous mode or after the previous interpolation time has expired in asynchronous mode.

An input buffer for interpolation data records eases the data exchange between control device and drive device. The real-time requirements to the network as well as to the drive device decrease in this case, because an input buffer decouples the data processing in the drive device from the data transmission on the network.

13.2.2 Linear interpolated position mode with several axes

In order to follow a two- or more-dimensional curve through the space with a defined speed, the control device calculates the different positions P_i for each set of coordinates which shall be reached at specified times t_i.

For each set-point P_i the control device shall calculate x_i, y_i .. and t_i. Each axis gets a set of interpolation data records, which each axis shall process internally independent from the other axes according to the chosen interpolation mode. This is shown in Figure 37.

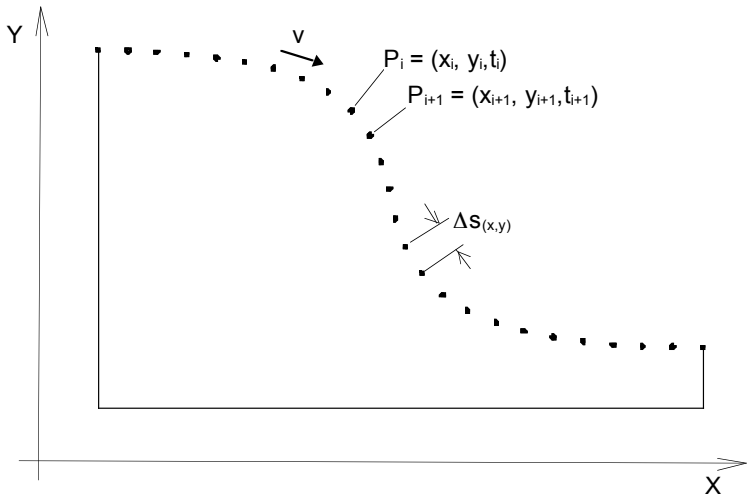


Figure 37 – Interpolated position mode for two axes

In a centralised drive system with a remote motion device doing the interpolation calculation, a central clocking scheme for synchronisation of the different axes. 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 axis. This is detailed in Table 155.

Table 155 – Position calculation in interpolated position mode for several axes

Calculated positions	*ip* data records for		
	x-axis	y-axis	z-axis
P_i	x_i, t_i	y_i, t_i	z_i, t_i
$P_i + 1$	$x_i + 1$, $t_i + 1$	$y_i + 1$, $t_i + 1$	$z_i + 1$, $t_i + 1$
$P_i + 2$	$x_i + 2$, $t_i + 2$	$y_i + 2$, $t_i + 2$	$z_i + 2$, $t_i + 2$
…	…	…	…
$P_i + n$	$x_i + n$, $t_i + n$	$y_i + n$, $t_i + n$	$z_i + n$, $t_i + n$

In decentralised motion systems, the control device 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 synchronises them. Each axis calculates internally and independently the necessary speed and acceleration needed to move from one position to the next. This may be done by calculating a linear or any other move between two given position set-points. Along this track, every axis controls the movement between the set-points independently from the other axes. The axes may continue their movement, 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 axis may act as shown in Figure 38.

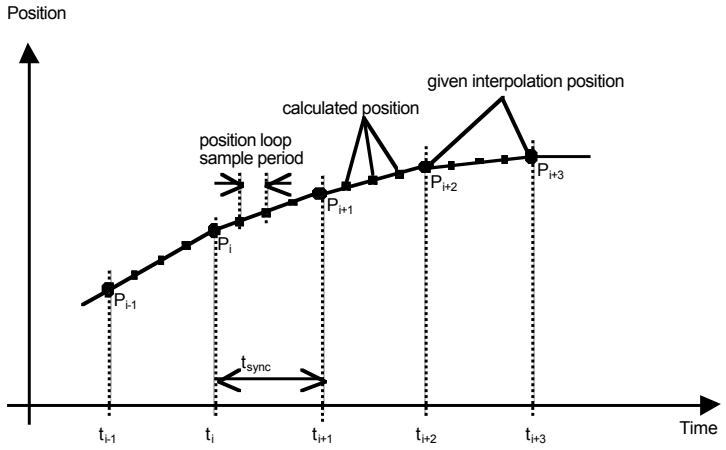


Figure 38 – Linear interpolation for one axis

NOTE In CANopen synchronous mode, the interpolation time is normally the same as the nominal period for the sync signal.

13.2.3 Buffer strategies for the interpolated position mode

If a drive device provides an input buffer for interpolation data records, its size may be organised by the control device using the interpolation data configuration. The control device 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 doesn't keep one complete data record, it should not be used. After the reorganisation of the input buffer, all previous stored data will be lost. All devices supporting the interpolated position mode shall implement an input buffer, which at least may keep one interpolation data record. The input buffer organisation is specified in Figure 39.

The content of the buffer items may 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 organised 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 control device may 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 a clear buffer, and after the reorganisation of the input buffer.

data record size # data record size # ::::: data record size # not accessible	parameter 1 parameter 2 ::::: parameter n parameter 1 parameter 2 ::::: parameter n ::::: parameter 1 parameter 2 ::::: parameter n	*ip data record* 1 *ip data record* 2 *ip data record* i	buffer size #

Figure 39 – Input buffer organisation

Figure 40 shows the difference between a FIFO buffer and a Ring buffer. The ring buffer may be used to achieve a periodic motion and all data records in the actual data buffer are considered to be valid. If no new data is written to the FIFO, then the motion shall halt and interpolation should become inactive at the last valid data point.

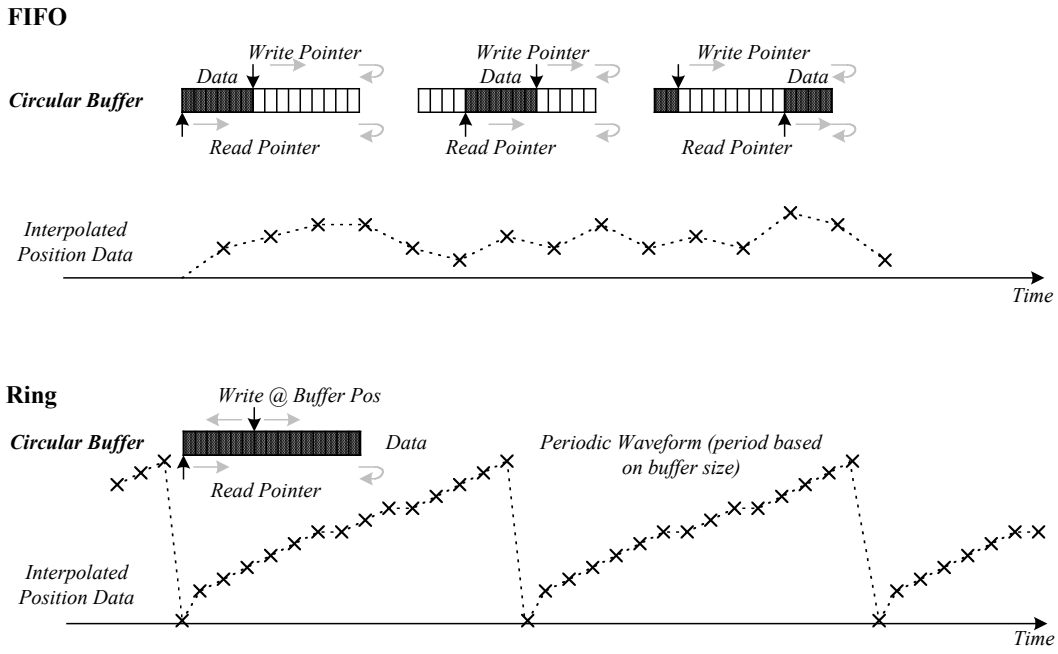


Figure 40 – Input buffer examples

13.2.4 Interpolated position mode FSA

Figure 41 specifies the interpolated position mode FSA. It is a sub FSA of the *Operation enable* state as shown in Figure 3.

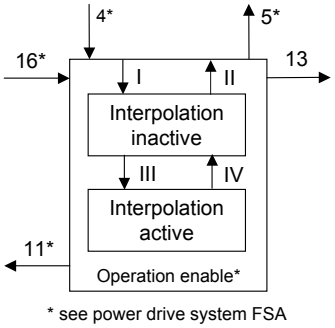


Figure 41 – Interpolated position mode FSA

The FSA states shall support the functions as shown in Table 156.

Table 156 – FSA states and supported functions

FSA state	Function
Interpolation inactive	The drive device will accept input data and will buffer it for interpolation calculations, but it does not move the axis.
Interpolation active	The drive unit will accept input data and it moves the axis.

The drive device supporting the *ip* mode shall support the transitions and actions as given in Table 157. The events shall initiate the transitions. The transition shall be terminated, after the action has been performed.

Table 157 – Transition events and actions

Transition	Event(s)	Action(s)
I	*ip* mode selected (see object 6060_h)	none
II	*ip* mode not selected (see object 6060_h)	none
III	Enable interpolation (bit 4 of the controlword is 1)	none
IV	Disable interpolation (bit 4 of the controlword is 0)	none

13.3 General definitions

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

13.4 Use of controlword and statusword

The interpolated position mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 42 shows the structure of the controlword. Table 158 defines the values for bit 4 and bit 8 of the controlword.

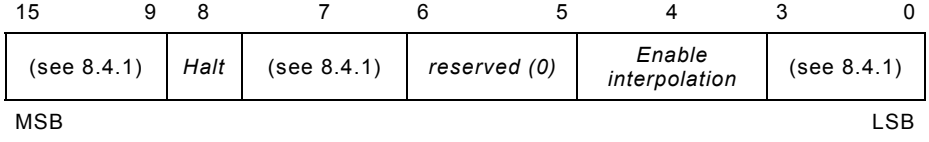


Figure 42 – Controlword for interpolated position mode

Table 158 – Definition of bit 4 and bit 8

Bit	Value	Definition
4	0	Disable *interpolation*
	1	Enable *interpolation*
8	0	Execute instruction of bit 4
	1	Axis shall be stopped accordingly to halt option code ($605D_h$), and bit 12 in the statusword shall be set to 0

Figure 43 shows the structure of the statusword. Table 159 defines the values for bit 10 and bit 12 of the statusword. The *target position reached* bit shall remain 0 until all set-points are processed.

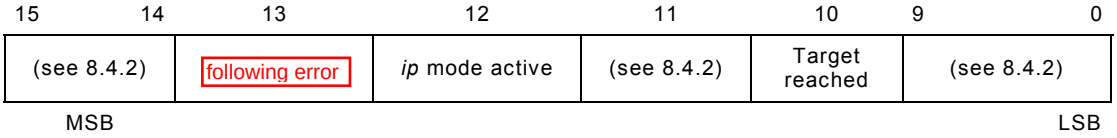


Figure 43 – Statusword for interpolated position mode

Table 159 – Definition of bit 10 and bit 12 and bit13

Bit	Value	Definition
10	0	Target position not (yet) reached (if Halt bit in last controlword was 0) or axle decelerates (if Halt bit in last controlword was 1)
	1	Target position reached (if Halt bit in last controlword was 0) or axle has veloxity 0 (if halt bit in last controlword was 1)
12	0	*Interpolation* inactive
	1	*Interpolation* active
13	0	No following error
	1	Following error

13.5 Detailed object definitions

13.5.1 Object $60C0_h$: Interpolation sub mode select

This object shall indicate the actually chosen interpolation mode. If linear interpolation is the only algorithm avalaible, then it is not necessary to implement this object. If a manufacturer-specific interpolation mode is selected, the corresponding interpolation data record shall be implemented in the manufacturer-specific profile area of the object dictionary. If the linear interpolation mode is selected, the interpolation data given in object $60C1_h$ shall be used. Table 160 specifies the value definition, Table 161 specifies the object description, and Table 162 specifies the entry description.

Table 160 – Value definition

Value	Definition
-32 768 to -1	Manufacturer-specific
0	Linear interpolation
+1 to +32 767	Reserved

Table 161 – Object description

Attribute	Value
Index	$60C0_h$
Name	Interpolation sub mode select
Object code	Variable
Data type	Integer16
Category	Optional

Table 162 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 160
Default value	0

13.5.2 Object $60C1_h$: Interpolation data record

This object shall indicate 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 is regarded as a new position set-point. To describe a cubic spline interpolation, 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 drive device's input buffer, the pointer of the buffer shall be automatically incremented to the next buffer position.

Table 163 specifies the object description, and Table 164 specifies the entry description.

Table 163 – Object description

Attribute	Value
Index	$60C1_h$
Name	~~Interpolated data record~~ Interpolation data record
Object code	Array
Data type	Integer32
Category	Optional

Table 164 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	01_h to FE_h
Default value	~~No~~ Manufacturer-specific
Sub-Index	01_h
Description	1st set-point
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	2nd set-point
Entry Category	Optional
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific
to	
Sub-Index	FE_h
Description	254th set-point
Entry Category	Optional
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

13.5.3 Object $60C2_h$: Interpolation time period

This object shall indicate the configured interpolation cycle time. The interpolation time period (sub-index 01_h) value shall be given in $10^{(\text{interpolation time index})}$ s(econd). The interpolation time index (sub-index 02_h) shall be dimensionless.

Table 165 specifies the object description, and Table 166 specifies the entry description.

Table 165 – Object description

Attribute	Value
Index	$60C2_h$
Name	Interpolation time period
Object code	Record
Data type	Interpolation time period record (0080_h)
Category	Conditional: mandatory if *ip, csp, csv* or *cst* mode is supported

Table 166 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	Interpolation time period value
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned8
Default value	01_h
Sub-Index	02_h
Description	Interpolation time index
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	-128 to +63
Default value	-3

13.5.4 Object $60C4_h$: Interpolation data configuration

This object shall provide the maximum buffer size, shall indicate the configured buffer organisation of interpolation data, and shall provide objects to define the size of the data record and to clear the buffers. This object is used to enable the drive device to receive the needed data in advance. It also is used to store the positions and further data sent by the control device.

The value of sub-index 01_h shall be given in number of interpolation data records.

The value of sub-index 02_h shall be given in number of interpolation data records.

If sub-index 03_h is 00_h this shall indicate a FIFO buffer oganisation, if it is 01_h this shall indicate a ring buffer organisation. All other values are reserved.

The value of sub-index 04_h shall be dimensionless indicating the next free buffer entry point.

The value of sub-index 05_h shall be given in byte.

If 00_h is written to sub-index 06_h this shall clear the buffer inputs, shall disable the access, and shall clear all *ip* data records. If 01_h is written to sub-index 06_h, this enables access to the input buffers. All other values are reserved.

Table 167 specifies the object description, and Table 168 specifies the entry description.

Table 167 – Object description

Attribute	Value
Index	$60C4_h$
Name	Interpolation data configuration
Object code	Record
Data type	Interpolation data configuration record (0081_h)
Category	Optional

Table 168 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	07_h
Default value	07_h
Sub-Index	01_h
Description	Maximum buffer size
Entry Category	Mandatory
Access	ro
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	No

Attribute	Value
Sub-Index	02_h
Description	Actual buffer size
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	0000 0000_h
Sub-Index	03_h
Description	Buffer organisation
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	00_h or 01_h
Default value	00_h
Sub-Index	04_h
Description	Buffer position
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	0000_h
Sub-Index	05_h
Description	Size of data record
Entry Category	Mandatory
Access	wo
PDO mapping	See /CiA402-3/
Value range	01_h to FE_h
Default value	01_h
Sub-Index	06_h
Description	Buffer clear
Entry Category	Mandatory
Access	wo
PDO mapping	See /CiA402-3/
Value range	00_h or 01_h
Default value	00_h

14 Profile velocity mode

14.1 General information

The profile velocity mode covers 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
- 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 includes and are described in Clause 10:

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

Various sensors may be used for velocity capture. In particular, the aim is that costs are reduced and the drive power system is simplified by evaluating position and velocity using a common sensor, such as is optional 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.~~

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

14.2 Functional description

Figure 44 shows the defined structure of the profile velocity mode. The actual velocity may be obtained through differentiation from the position encoder and is represented in position encoder increments.

The *target reached* bit (bit 10) shall be set to 1 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.

As soon as the velocity actual value exceeds the velocity threshold longer than the *velocity threshold time*, then bit 12 shall be set to 0 in the statusword. Below this threshold the bit shall be set to 1 and shall indicate that the axis is stationary.

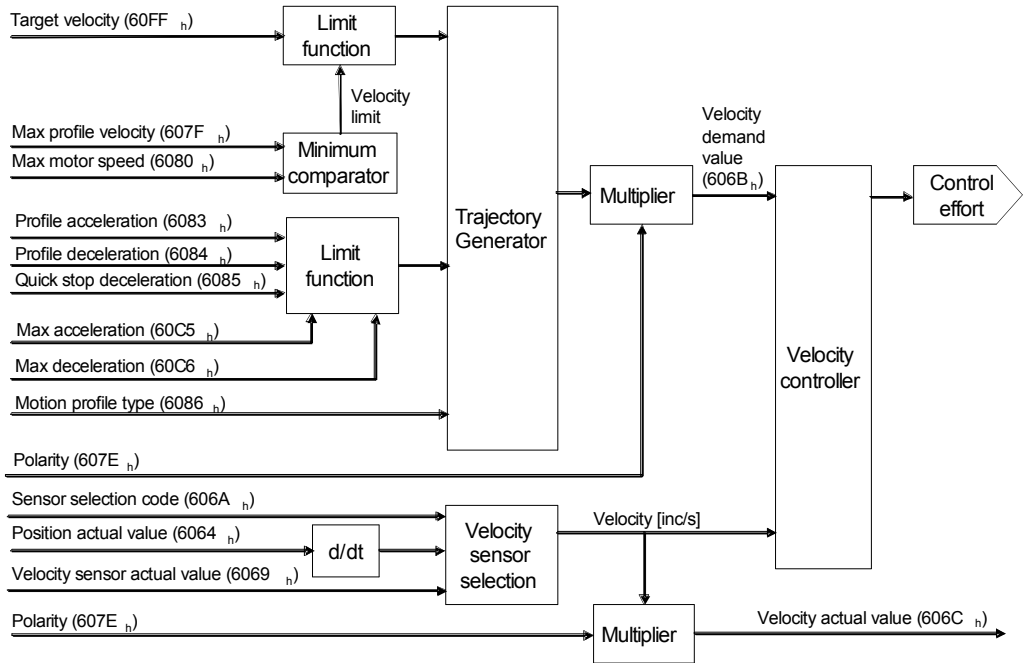


Figure 44 – Profile velocity mode

14.3 General definitions

The factors necessary for scaling have a linear relationship and therefore they are described in the factor group. The polarity is described in the factor group as well.

14.4 Use of controlword and statusword

The profile velocity mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 45 shows the structure of the controlword. Table 169 defines the values for bit 8 of the controlword.

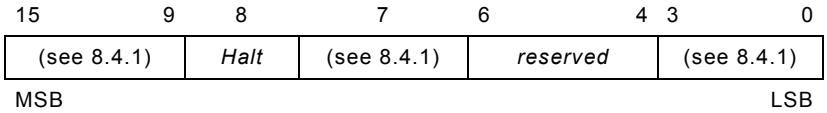


Figure 45 – Controlword for profile velocity mode

Table 169 – Definition of bit 8

Bit	Value	Definition
8	0	The motion shall be executed or continued
	1	Axis shall be stopped according to the halt option code ($605D_h$)

Figure 46 shows the structure of the statusword. Table 170 defines the values for bit 10, 12, and 13 of the statusword.

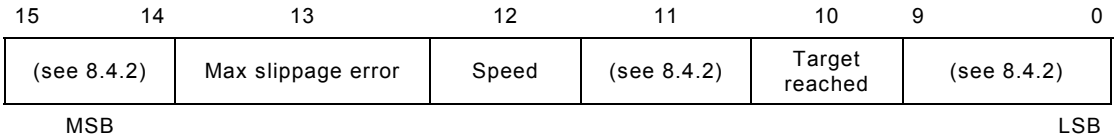


Figure 46 – Statusword for profile velocity mode

Table 170 – Definition of bit 10, bit 12, and bit 13

Bit	Value	Definition
10	0	Halt (Bit 8 in controlword) = 0: Target not reached Halt (Bit 8 in controlword) = 1: Axis decelerates
	1	Halt (Bit 8 in controlword) = 0: Target reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0
12	0	Speed is not equal 0
	1	Speed is equal 0
13	0	Maximum slippage not reached
	1	Maximum slippage reached

14.5 Detailed object definitions

14.5.1 Object 6069_h: Velocity sensor actual value

This object shall provide the value read from a velocity sensor. The value shall be given in increments per second. Table 171 specifies the object description, and Table 172 specifies the entry description.

Table 171 – Object description

Attribute	Value
Index	6069_h
Name	Velocity sensor actual value
Object code	Variable
Data type	Integer32
Category	Optional

Table 172 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

14.5.2 Object $606A_h$: Sensor selection code

This object shall provide the source of the velocity sensor actual value. It determines whether a differentiated position signal or the signal from a separate velocity sensor is evaluated.

Table 173 specifies the value definition, Table 174 specifies the object description, and Table 175 specifies the entry description.

Table 173 – Value definition

Value	Definition
0000_h	Actual velocity value from position encoder
0001_h	Actual velocity value from velocity encoder
0002_h to $7FFF_h$	Reserved
8000_h to $FFFF_h$	Manufacturer-specific

Table 174 – Object description

Attribute	Value
Index	$606A_h$
Name	Sensor selection code
Object code	Variable
Data type	Integer16
Category	Optional

Table 175 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	See Table 173
Default value	Manufacturer-specific

14.5.3 Object $606B_h$: Velocity demand value

This object shall provide the output value of the trajectory generator. The value shall be given in the user-defined velocity units. Table 176 specifies the object description, and Table 177 specifies the entry description.

Table 176 – Object description

Attribute	Value
Index	$606B_h$
Name	Velocity demand value
Object code	Variable
Data type	Integer32
Category	Optional

Table 177 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

14.5.4 Object $606C_h$: Velocity actual value

This object shall provide the actual velocity value derived either from the velocity sensor or the position sensor. The value shall be given in user-defined velocity units. Table 178 specifies the object description, and Table 179 specifies the entry description.

Table 178 – Object description

Attribute	Value
Index	$606C_h$
Name	Velocity actual value
Object code	Variable
Data type	Integer32
Category	Conditional: mandatory if *pv* or *csv* is supported

Table 179 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	No

14.5.5 Object $606D_h$: Velocity window

This object shall indicate the configured velocity window. The value shall be given in user-defined velocity units. Table 180 specifies the object description, and Table 181 specifies the entry description.

Table 180 – Object description

Attribute	Value
Index	$606D_h$
Name	Velocity window
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 181 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

14.5.6 Object $606E_h$: Velocity window time

This object shall indicate the configured velocity window time. The value shall be given in milliseconds. Table 182 specifies the object description, and Table 183 specifies the entry description.

Table 182 – Object description

Attribute	Value
Index	$606E_h$
Name	Velocity window time
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 183 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	0000_h

14.5.7 Object $606F_h$: Velocity threshold

This object shall indicate the configured velocity threshold. The value shall be given in user-defined velocity units. Table 184 specifies the object description, and Table 185 specifies the entry description.

Table 184 – Object description

Attribute	Value
Index	$606F_h$
Name	Velocity threshold
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 185 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

14.5.8 Object 6070_h: Velocity threshold time

This object shall indicate the configured velocity threshold time. The value shall be given in milliseconds. Table 186 specifies the object description, and Table 187 specifies the entry description.

Table 186 – Object description

Attribute	Value
Index	6070_h
Name	Velocity threshold time
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 187 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

14.5.9 Object $60FF_h$: Target velocity

This object shall indicate the configured target velocity and shall be used as input for the trajectory generator. The value shall be given in user-defined velocity units. Table 188 specifies the object description, and Table 189 specifies the entry description.

Table 188 – Object description

Attribute	Value
Index	$60FF_h$
Name	Target velocity
Object code	Variable
Data type	Integer32
Category	Conditional: mandatory if *pv* or *csv* is supported

Table 189 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

14.5.10 Object $60F8_h$: Max slippage

This object shall indicate the configured maximal slippage of an asynchronous motor. When the max slippage has been reached, the corresponding bit 13 max slippage error in the statusword shall be set to 1. The reaction of the drive device, when the max slippage error occurs, is manufacturer-specific. This value shall be given in user-defined units. Table 190 specifies the object description, and Table 191 specifies the entry description.

Table 190 – Object description

Attribute	Value
Index	$60F8_h$
Name	Max slippage
Object code	Variable
Data type	Integer32
Category	Optional

Table 191 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	Manufacturer-specific

15 Profile torque mode

15.1 General information

The profile torque mode allows control device (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.

15.2 Functional description

If the control device switches 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 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 are not modified. As an example, the linear motor target force is 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 drive profile specification. 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 may be handled (i.e. downloaded) in a standard way. Their detailed data definition is manufacturer-specific.

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

Figure 47 shows the defined structure of the profile torque mode.

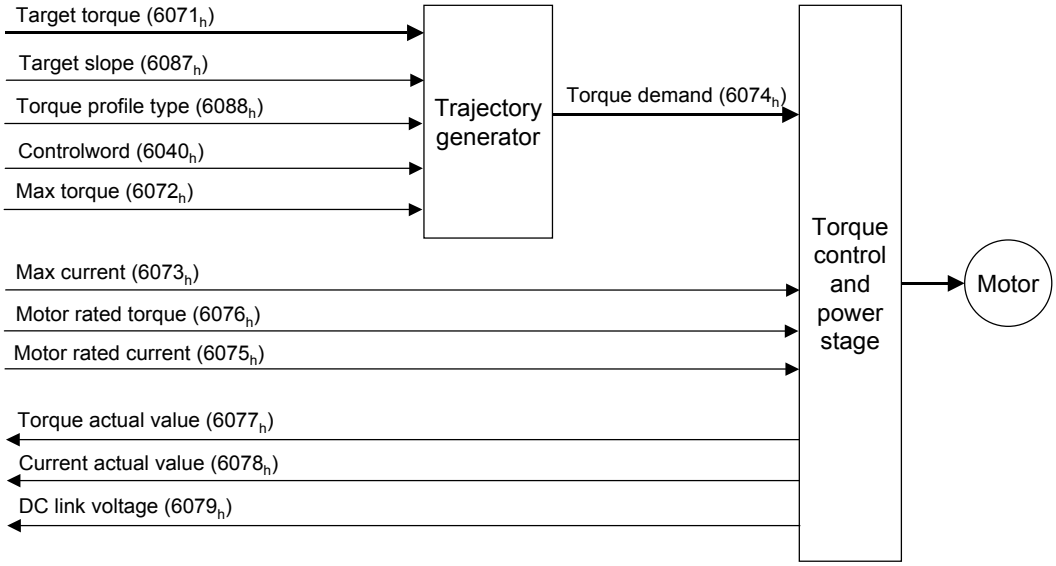


Figure 47 – Structure of the profile torque mode

15.3 General definitions

There are no general definitions given for the profile torque mode.

15.4 Use of controlword and statusword

The profile torque mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 48 shows the structure of the controlword. Table 192 defines the values for bit 8 of the controlword.

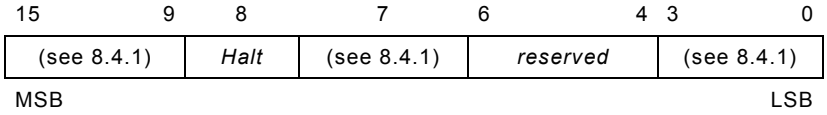


Figure 48 – Controlword for profile torque mode

Table 192 – Definition of bit 8

Bit	Value	Definition
8	0	The motion shall be executed or continued
	1	Axis shall be stopped according to the halt option code ($605D_h$)

Figure 49 shows the structure of the statusword. Table 193 defines the values for bit 10 of the statusword.

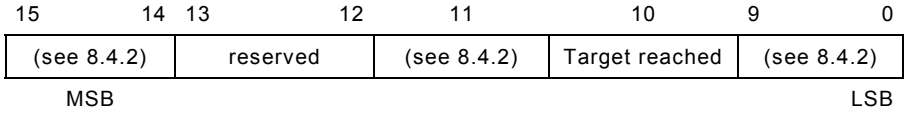


Figure 49 – Statusword for profile torque mode

Table 193 – Definition of bit 10

Bit	Value	Definition
10	0	Halt (Bit 8 in controlword) = 0: *Target torque* not reached Halt (Bit 8 in controlword) = 1: Axis decelerates
	1	Halt (Bit 8 in controlword) = 0: *Target torque* reached Halt (Bit 8 in controlword) = 1: Velocity of axis is 0

Note *Target torque* reached is defined by a manufacturer-specific time or window object.

15.5 Detailed object definitions

15.5.1 Object 6071_h: Target torque

This object shall indicate the configured input value for the torque controller in profile torque mode. The value shall be given per thousand of rated torque. Table 194 specifies the object description, and Table 195 specifies the entry description.

Table 194 – Object description

Attribute	Value
Index	6071_h
Name	Target torque
Object code	Variable
Data type	Integer16
Category	Conditional: mandatory if *tq* or *cst* is supported

Table 195 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	0000_h

15.5.2 Object 6072_h: Max torque

This object shall indicate the configured maximum permissible torque in the motor. The value shall be given per thousand of rated torque. Table 196 specifies the object description, and Table 197 specifies the entry description.

Table 196 – Object description

Attribute	Value
Index	6072_h
Name	Max torque
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 197 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

15.5.3 Object 6073_h: Max current

This object shall indicate the configured maximum permissible torque creating current in the motor. The value shall be given per thousand of rated current. Table 198 specifies the object description, and Table 199 specifies the entry description.

Table 198 – Object description

Attribute	Value
Index	6073_h
Name	Max current
Object code	Variable
Data type	Unsigned16
Category	Optional

Table 199 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

15.5.4 Object 6074_h: Torque demand

This object shall provide the output value of the trajectory generator. The value shall be given in 1/1 000 of rated torque. Table 200 specifies the object description, and Table 201 specifies the entry description.

Table 200 – Object description

Attribute	Value
Index	6074_h
Name	Torque demand
Object code	Variable
Data type	Integer16
Category	Optional

Table 201 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	No

15.5.5 Object 6075_h: Motor rated current

This object shall indicate the configured motor rated current. It is taken from the motor's name-plate. Depending on the motor and drive technology this current is DC, peak or r.m.s. (root-mean-square) current. All relative current data refers to this value. The value shall be given in mA. Table 202 specifies the object description, and Table 203 specifies the entry description.

Table 202 – Object description

Attribute	Value
Index	6075_h
Name	Motor rated current
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 203 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

15.5.6 Object 6076_h: Motor rated torque

This object shall indicate the configured motor rated torque. It is taken from the motor's name-plate. All relative torque data shall refer to this value. For linear motors, the object name is not changed, but the motor rated force value shall be entered as multiples of mN (milli Newton). The value shall be given in mNm (milli Newton metre). Table 204 specifies the object description, and Table 205 specifies the entry description.

Table 204 – Object description

Attribute	Value
Index	6076_h
Name	Motor rated torque
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 205 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

15.5.7 Object 6077_h: Torque actual value

This object shall provide the actual value of the torque. It shall correspond to the instantaneous torque in the motor. The value shall be given per thousand of rated torque. Table 206 specifies the object description, and Table 207 specifies the entry description.

Table 206 – Object description

Attribute	Value
Index	6077_h
Name	Torque actual value
Object code	Variable
Data type	Integer16
Category	Conditional: mandatory if *cst* is supported

Table 207 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	No

15.5.8 Object 6078_h: Current actual value

This object shall provide the actual value of the current. It shall correspond to the current in the motor. The value shall be given per thousand of rated current. Table 208 specifies the object description, and Table 209 specifies the entry description.

Table 208 – Object description

Attribute	Value
Index	6078_h
Name	Current actual value
Object code	Variable
Data type	Integer16
Category	Optional

Table 209 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	No

15.5.9 Object 6079_h: DC link circuit voltage

This object shall provide the instantaneous DC link current voltage at the drive device. The value shall be given in mV. Table 210 specifies the object description, and Table 211 specifies the entry description.

Table 210 – Object description

Attribute	Value
Index	6079_h
Name	DC link circuit voltage
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 211 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	No

15.5.10 Object 6087_h: Torque slope

This object shall indicate the configured rate of change of torque. The value shall be given in units of per thousand of rated torque per second. Table 212 specifies the object description, and Table 213 specifies the entry description.

Table 212 – Object description

Attribute	Value
Index	6087_h
Name	Torque slope
Object code	Variable
Data type	Unsigned32
Category	Conditional: mandatory if *tq* is supported

Table 213 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

15.5.11 Object 6088_h: Torque profile type

This object shall indicate the configured type of profile used to perform a torque change. Table 214 specifies the value definition, Table 215 specifies the object description, and Table 216 specifies the entry description.

Table 214 – Value definition

Value	Definition
0000_h	Linear ramp (trapezoidal profile)
0001_h	$\sin^2$ ramp
0002_h to $7FFF_h$	Reserved
8000_h to $FFFF_h$	Manfacturer-specific

Table 215 – Object description

Attribute	Value
Index	6088_h
Name	Torque profile type
Object code	Variable
Data type	Integer16
Category	Optional

Table 216 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	0000_h

16 Velocity mode

16.1 General information

This mode is used by frequency inverters but not limited to this kind of drive device. Most applications use a velocity set-point and a controlword for switching the drive device on and off.

Figure 50 shows the overall structure of the velocity mode. The possible torque control function is not in the scope of this part of profile specification, it may use the target torque and torque actual value objects defined in 15.5.1 or respectively in 15.5.7.

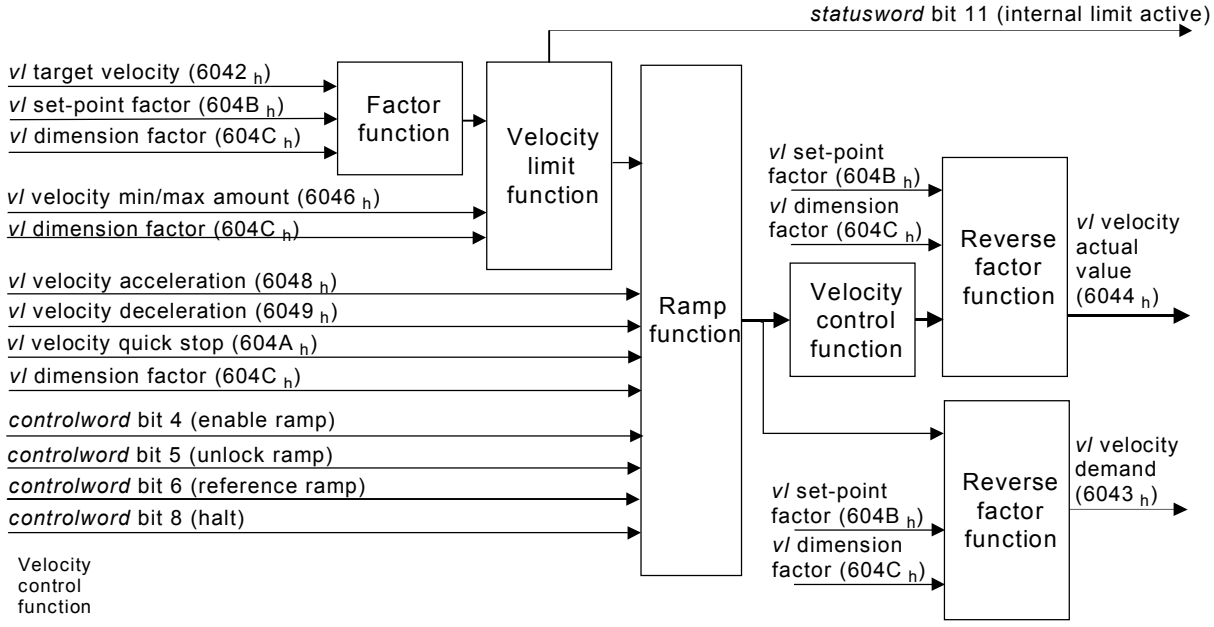


Figure 50 – Velocity mode with all objects

All drive devices using this profile and supporting the velocity mode shall implement the mandatory objects and there functionality as shown in Figure 51.

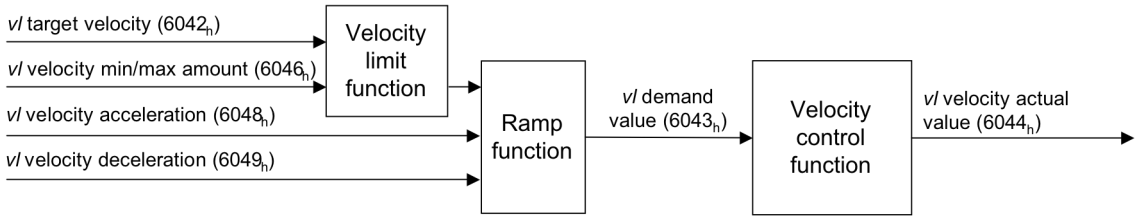


Figure 51 – Velocity mode with mandatory objects only

16.2 Functional description

16.2.1 Velocity limit function

The limits in the velocity limit function may be given in user-specific units by including the *vl* dimension factor in the velocity limit or in rotations per minute (rpm). 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.

16.2.2 Ramp function

Figure 52 shows the velocity profile that 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 as defined in *vl* velocity acceleration, *vl* velocity deceleration, and *vl* velocity quickstop.

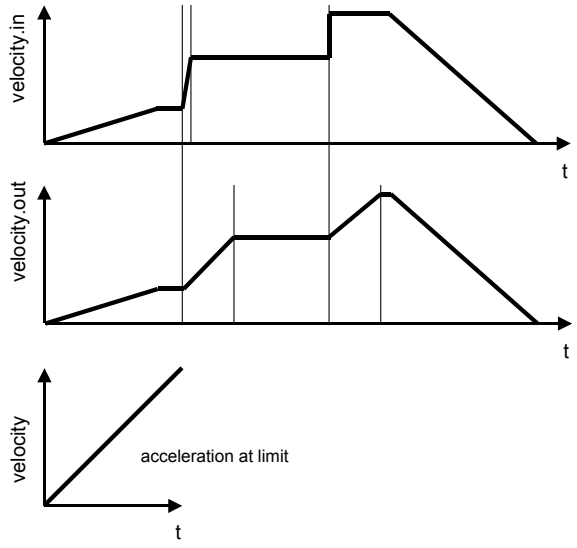


Figure 52 – Velocity profile

16.2.3 Velocity control function

On the basis of the *vl* velocity demand, the velocity control function provides the *vl* control-effort.

16.2.4 Factor function

The factor function multiplies the input variables by the assigned factors. The factor shall have a value of 1, if it is not implemented.

Figure 53 shows the structure of the factor function; the factor function for two factors is built of two functions in series connection.

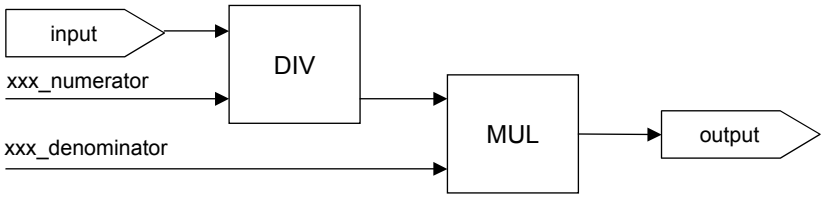


Figure 53 – Factor function

Figure 54 shows the structure of the reverse factor function. The reverse factor function divides the input variables by the assigned factors.

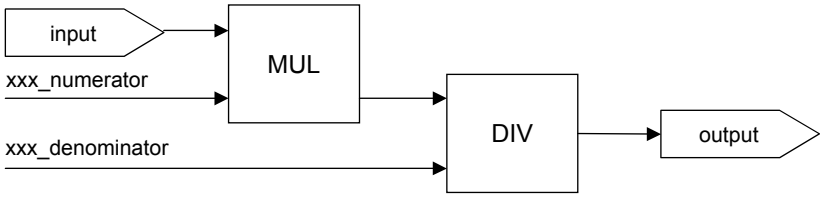


Figure 54 – Reverse factor function

16.3 General definitions

All objects defined in 16.5 are used only for the velocity mode.

16.4 Use of controlword and statusword

The velocity mode uses some bits of the controlword and the statusword for mode-specific purposes. Figure 55 shows the structure of the controlword. Table 217 and Figure 56 define the values for bit 4, bit 5, bit 6, and bit 8 of the controlword. These bits are optional.

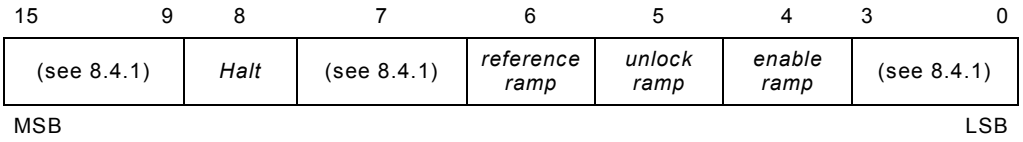


Figure 55 – ~~Controlword for profile velocity mode~~
Controlword for velocity mode

Table 217 – Definition of bit 4, bit 5, bit 6, and bit 8

Bit	Value	Definition
4	0	Velocity demand value shall be controlled in any other (manufacturer-specific) way, for example by a test function generator or manufacturer-specific halt function
	1	Velocity demand value shall accord with ramp output value
5	0	Ramp output value shall be locked to current output value
	1	Ramp output value shall follow ramp input value
6	0	Ramp input value shall be set to zero
	1	Ramp input value shall accord with ramp reference
8	0	No comand
	1	Motor shall be stopped

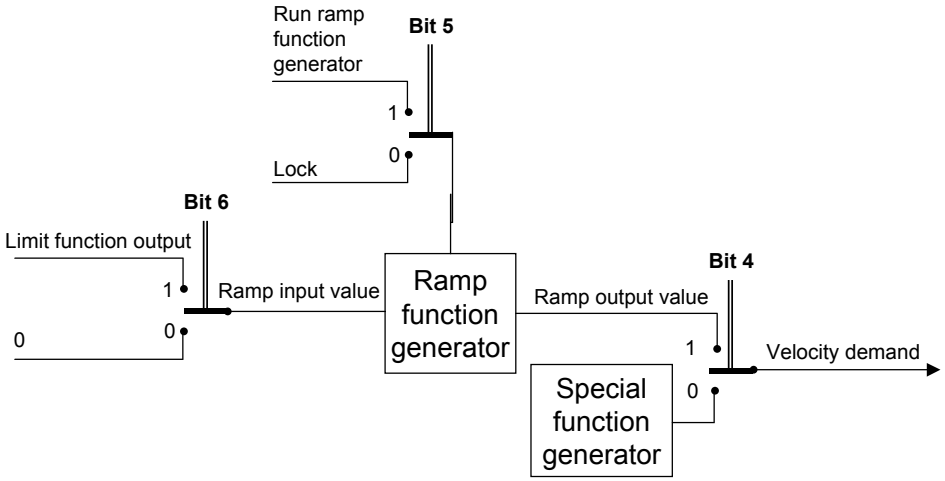


Figure 56 – Usage of controlword bits in velocity mode

Figure 57 shows the structure of the statusword.

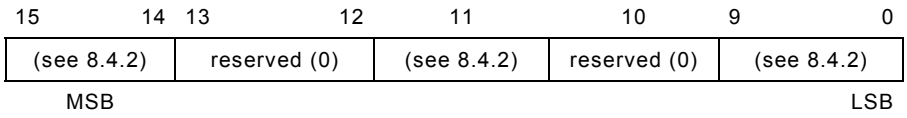


Figure 57 – ~~Statusword for profile velocity mode~~
Statusword for velocity mode

16.5 Detailed object definitions

16.5.1 Object 6042_h: *vl* target velocity

This object shall indicate the required velocity of the system. It shall be multiplied by the *vl* dimension factor and the *vl* set-point factor, if these are implemented. The value shall be given in user-defined velocity units or in revolutions per minute (rpm), if the *vl* dimension factor and the *vl* set-point factor are not implemented or have the value 1. Positive values shall indicate forward direction and negative values shall indicate reverse direction. Table 218 specifies the object description, and Table 219 specifies the entry description.

Table 218 – Object description

Attribute	Value
Index	6042_h
Name	*vl* target velocity
Object code	Variable
Data type	Integer16
Category	Conditional: mandatory if *vl* is supported

Table 219 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	0000_h

16.5.2 Object 6043_h: *vl* velocity demand

This object shall provide the instantaneous velocity generated by the ramp function. It is an internal object of the drive device. The value shall be given in the very same unit as the *vl* target velocity. Positive values shall indicate forward direction and negative values shall indicate reverse direction. Table 220 specifies the object description, and Table 221 specifies the entry description.

Table 220 – Object description

Attribute	Value
Index	6043_h
Name	*vl* velocity demand
Object code	Variable
Data type	Integer16
Category	Conditional: mandatory if *vl* is supported

Table 221 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	No

16.5.3 Object 6044_h: *vl* velocity actual value

This object shall provide the velocity at the motor spindle or load. Depending on the implementation (simple drive device, without sensor, with sensor, etc.), the drive shall provide the appropriate image of the actual velocity ~~(velocity demand, velocity control effort, calculated velocity, measured velocity).~~ Replace: derived for example velocity demand or a sensor signal.

The value shall be given in the very same unit as the *vl* target velocity. Positive values shall indicate forward direction and negative values shall indicate reverse direction. Table 222 specifies the object description, and Table 223 specifies the entry description.

Table 222 – Object description

Attribute	Value
Index	6044_h
Name	*vl* velocity actual value
Object code	Variable
Data type	Integer16
Category	Conditional: mandatory, if *vl* is supported

Table 223 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	No

16.5.4 Object 6046_h: *vl* velocity min max amount

This object shall indicate the configured minimum and maximum amount of velocity. The *vl* velocity max amount sub-object shall be mapped internally to the *vl* velocity max pos and *vl* velocity max neg values. The *vl* velocity min amount sub-object shall be mapped internally to the *vl* velocity min pos and *vl* velocity min neg values.

This transfer characteristic is shown in Figure 58.

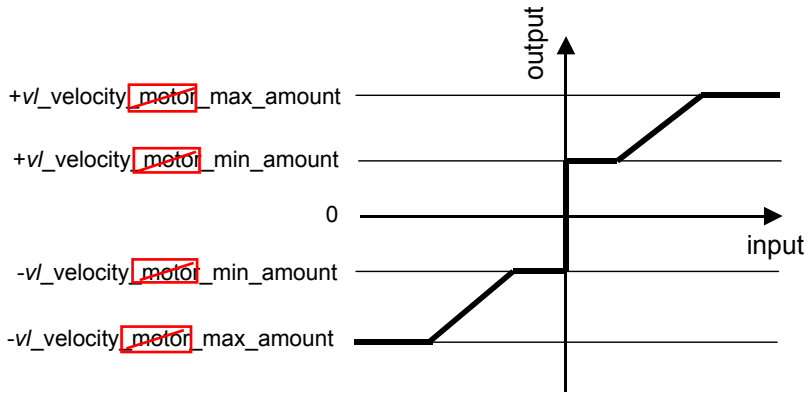


Figure 58 – Transfer characteristic of *vl* velocity min max amount

The values shall be given in rotations per minute (rpm) or in user-defined velocity unit if the vl dimension factor object is implemented and is not set to 1. Table 224 specifies the object description, and Table 225 specifies the entry description.

Table 224 – Object description

Attribute	Value
Index	6046_h
Name	*vl* velocity min max amount
Object code	Array
Data type	Unsigned32
Category	Conditional: mandatory if *vl* mode is supported

Table 225 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	*vl* velocity min amount
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	*vl* velocity max amount
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

16.5.5 Object 6049_h: *vl* velocity deceleration

This object shall indicate the configured delta speed and delta time of the slope of the deceleration ramp as shown in Figure 59.

$$vl\ velocity\ deceleration = \frac{delta\ speed}{delta\ time}$$

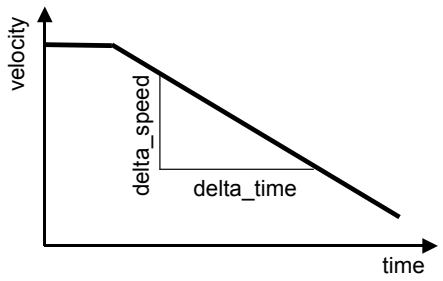


Figure 59 – Transfer characteristic of the velocity deceleration

The value of delta speed shall be given in rotations per minute (rpm) or in a user-defined velocity unit if the *vl* dimension factor object is implemented and is not set to 1; the value of delta time shall be given in s. Table 226 specifies the object description, and Table 227 specifies the entry description. If this object is not implemented, the value in object 6048_h shall be used for *vl velocity deceleration*.

Table 226 – Object description

Attribute	Value
Index	6049_h
Name	*vl* velocity deceleration
Object code	Record
Data type	*vl* velocity acceleration decelaration
Category	~~Conditional~~: optional

Table 227 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific

Attribute	Value
Sub-Index	02_h
Description	Delta time
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

16.5.6 Object 6048_h: *vl* velocity acceleration

This object shall indicate the configured delta speed and delta time of the slope of the acceleration ramp as shown in Figure 60.

Example: If you ramp to 1 500 rpm in 3,7 s, the delta speed equals to 15 000 rpm and delta time equals to 37 s.

$$vl\ velocity\ acceleration = \frac{delta\ speed}{delta\ time}$$

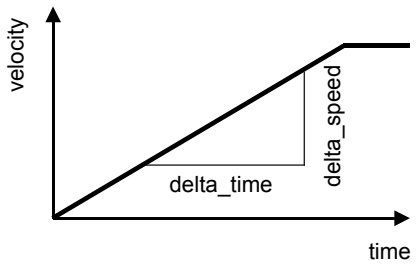


Figure 60 – Transfer characteristic of the velocity acceleration

The value of delta speed shall be given in rotations per minute (rpm) or in user-defined velocity unit if the *vl* dimension factor object is implemented and is not set to 1; the value of delta time shall be given in s. Table 228 specifies the object description, and Table 229 specifies the entry description.

Table 228 – Object description

Attribute	Value
Index	6048_h
Name	*vl* velocity acceleration
Object code	Record
Data type	*vl* velocity acceleration decelaration
Category	Conditional: mandatory if *vl* is supported

Table 229 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	Delta time
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

16.5.7 Object $604A_h$: *vl* velocity quick stop

This object shall indicate the configured delta speed and delta time of the slope of the deceleration ramp for quick stop as shown in Figure 61.

$$velocity\ quick\ stop = \frac{delta\ speed}{delta\ time}$$

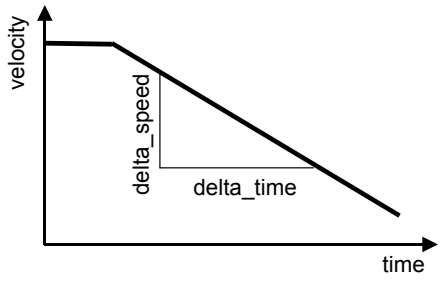


Figure 61 – Transfer characteristic of the quick stop deceleration

The value of delta speed shall be given in rotations per minute (rpm) or in user-defined velocity unit if the *vl* dimension factor object is implemented and is not set to 1; the value of

delta time shall be given in s. Table 230 specifies the object description, and Table 231 specifies the entry description.

Table 230 – Object description

Attribute	Value
Index	$604A_h$
Name	*v*/ velocity quick stop
Object code	Record
Data type	*v*/ velocity acceleration decelaration
Category	~~Conditional~~: optional

Table 231 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	Delta speed
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	Manufacturer-specific
Sub-Index	02_h
Description	Delta time
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned16
Default value	Manufacturer-specific

16.5.8 Object $604B_h$: *v*/ set-point factor

This object shall indicate the configured numerator and denominator of the *v*/ set-point factor. The *v*/ set-point factor serves to modify the resolution or directing range of the specified set-point. It is aso included in calculation of the *v*/ velocity demand, and *v*/ velocity actual value. It does not influence the velocity limit function and the ramp function. The value shall have no physical unit and shall be given in the range from -32 768 to +32 767, but the value of 0 shall

not be used. Table 232 specifies the object description, and Table 233 specifies the entry description.

Table 232 – Object description

Attribute	Value
Index	$604B_h$
Name	*v*/ set-point factor
Object code	Array
Data type	Integer16
Category	Optional

Table 233 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	*v*/ set-point factor numerator
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	See value definition
Default value	+1
Sub-Index	02_h
Description	*v*/ set-point factor denominator
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	See value definition
Default value	+1

16.5.9 Object $604C_h$: *v*/ dimension factor

This object shall indicate the configured numerator and denominator of the *v*/ dimension factor. The *v*/ dimension factor serves to include gearing in calculation or serves to scale the frequencies or specific units of the user. It influences the *v*/ target velocity, *v*/ velocity demand, *v*/ velocity actual value as well as the velocity limit function and the ramp function.

Calculating the *vl* dimension factor: Every user-specific velocity consists of a specific unit referred to a specific unit of time (e.g. 1/s, bottles/min, m/s, etc.). The purpose of the *vl* dimension factor is to convert this specific unit to the revolutions/minute unit.

Velocity [user-defined unit] × Dimension factor [rpm/user-defined unit] = Velocity [rpm]

The values shall be in the range of -2 147 483 648 to +2 147 483 647, but the value of 0 shall be not used.

Table 234 specifies the object description, and Table 235 specifies the entry description.

Example: If the target unit is 0,1 Hz the numerator is 120 and the denominator is the pole number.

Table 234 – Object description

Attribute	Value
Index	$604C_h$
Name	*vl* dimension factor
Object code	Array
Data type	Integer32
Category	Optional

Table 235 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	02_h
Default value	02_h
Sub-Index	01_h
Description	*vl* dimension factor numerator
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	See value definition
Default value	+1

Attribute	Value
Sub-Index	02_h
Description	*v*/ dimension factor denominator
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	See value definition
Default value	+1

17 Cyclic synchronous position mode

17.1 General information

The overall structure for this mode is shown in Figure 62. With this mode, the trajectory generator is located in the control device, not in the drive device. In cyclic synchronous manner, it provides a target position to the drive device, which performs position control, velocity control and torque control. Optionally, additive velocity and torque values can be provided by the control system in order to allow for velocity and/or torque feedforward. Measured by sensors, the drive device may provide actual values for position, velocity and torque to the control device.

The behavior of the control function is influenced by control parameters like limit functions, which are externally applicable. The drive internal control function is not specified more precisely in this part of profile specification as it is highly manufacturer-specific, but the format and content of the control parameters are provided.

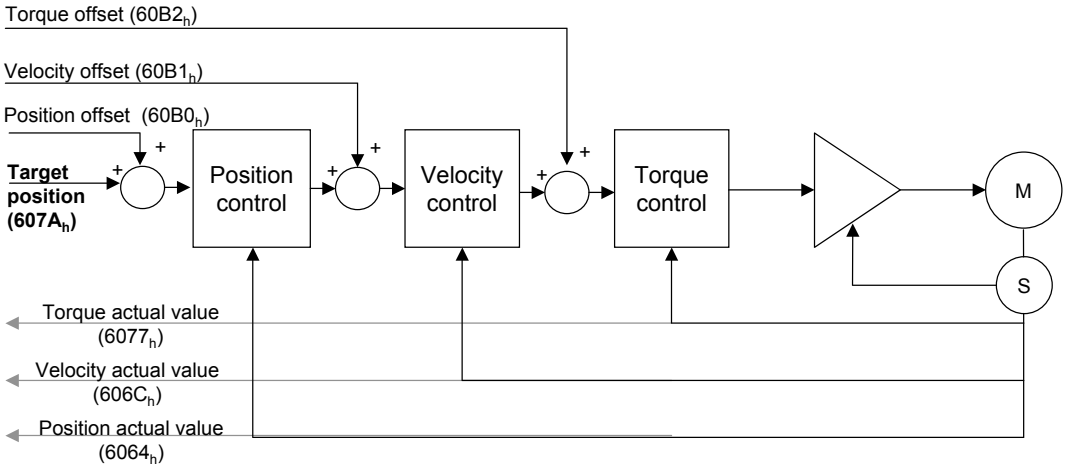


Figure 62 – Cyclic synchronous position mode overview

17.2 Functional description

Figure 63 shows the inputs and outputs of the drive control function. The input values (from the control function point of view) are the target position and optionally a position offset (to be added to the target position to allow two instances to set up the position) as well as an optional velocity offset and an optional torque offset used for feedforward control. Especially in cascaded control structures, where a position control is followed by a velocity or torque control, the output of the position control loop is used as an input for a further calculation in the drive device. Limit functions may be used to restrict the range of values to avoid unintended positions.

The drive device monitors the following error. Other features specified in this mode are limitation of motor speed and a quick stop function for emergency reasons. The torque may be limited as well.

The interpolation time period defines the time period between two updates of the target position and/or additive position and shall be used for intercycle interpolation.

The target position shall be interpreted as absolute value.

The position actual value is used as mandatory output to the control device. Further outputs may be the velocity actual value, torque actual value and the velocity sensor actual value. The following error actual value may be used as an additional parameter.

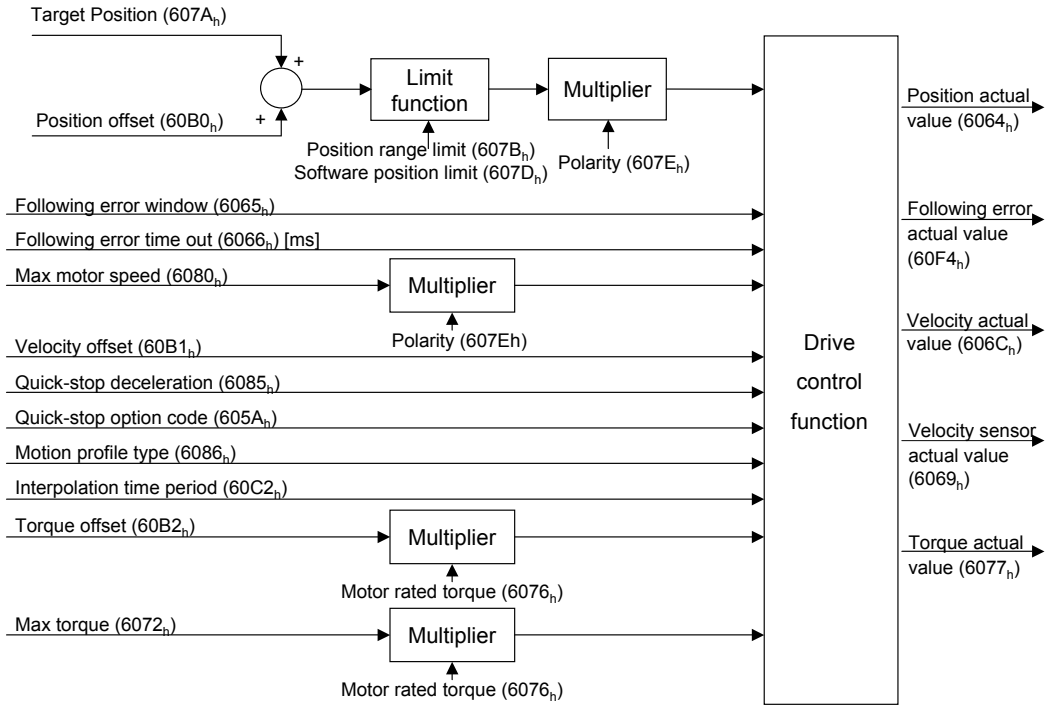


Figure 63 – Cyclic synchronous position control function

All values are transformed – if necessary – from user-defined units to normalised units such as increments with the functions described in Clause 9.

A *target position value* or *position offset* outside the allowed range of the *following error window* around a *position demand value* for longer than the *following error time out* shall result in setting bit 13 (*following error*) in the statusword to 1.

17.3 Use of controlword and statusword

The cyclic synchrounous position mode uses no mode specific bits of the controlword and three bits of the statusword for mode-specific purposes. Figure 64 shows the structure of the statusword. Table 236 defines the values for bit 10, 12, and 13 of the statusword.

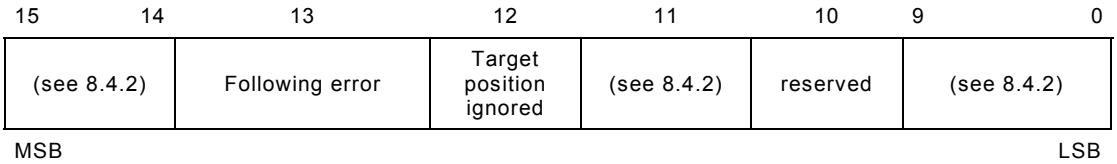


Figure 64 – Statusword for profile cyclic synchronous position mode

Table 236 – Definition of bit 10, bit 12, and bit 13

Bit	Value	Definition
10	0	Reserved
	1	Reserved
12	0	Target position ignored
	1	Target position shall be used as input to position control loop
13	0	No following error
	1	Following error

17.4 Detailed object definitions

17.4.1 Object $60B0_h$: Position offset

This object shall provide the offset of the target position. The offset shall be given in user-defined position units.

NOTE The value itself is absolute and thus independent of how often it is transmitted over the communication system, for example, transmitted twice does not mean double value. Since the additive position value represents an offset to the target position it can be also used to control the drive with relative values with regard to the target position.

Table 237 specifies the object description, and Table 238 specifies the entry description.

Table 237 – Object description

Attribute	Value
Index	$60B0_h$
Name	Position offset
Object code	Variable
Data type	Integer32
Category	Optional

Table 238 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	0

17.4.2 Object $60B1_h$: Velocity offset

This object shall provide the offset for the velocity value. The offset shall be given in user-defined velocity units. In cyclic synchronous position mode, this object contains the input value for velocity feed forward. In cyclic synchronous velocity mode (see Clause 18) it contains the commanded offset of the drive device.

NOTE The value itself is absolute and thus independent of how often it is transmitted over the communication system, for example transmitted twice does not mean double value. Since the additive velocity value represents an offset to the target velocity, it can be also used to control the drive with relative values with regard to the target velocity.

Table 239 specifies the object description, and Table 240 specifies the entry description.

Table 239 – Object description

Attribute	Value
Index	$60B1_h$
Name	Velocity offset
Object code	Variable
Data type	Integer32
Category	Optional

Table 240 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer32
Default value	0

17.4.3 Object $60B2_h$: Torque offset

This object shall provide the offset for the torque value. The offset shall be given in per thousand rated torque. In cyclic synchronous position mode and cyclic synchronous velocity mode (see Clause 18), this object contains the input value for torque feed forward. In cyclic synchronous torque mode (see Clause 18) it contains the commanded additive torque of the drive, which is added to the target torque value.

NOTE The value itself is absolute and thus independent of how often it is transmitted over the communication system, for example transmitted twice does not mean double value.

Table 241 specifies the object description, and Table 242 specifies the entry description.

Table 241 – Object description

Attribute	Value
Index	$60B2_h$
Name	Torque offset
Object code	Variable
Data type	Integer16
Category	Optional

Table 242 – Entry description

Attribute	Value
Sub-Index	00_h
Access	rw
PDO mapping	See /CiA402-3/
Value range	Integer16
Default value	0

18 Cyclic synchronous velocity mode

18.1 General information

The overall structure for this mode is shown in Figure 65. With this mode, the trajectory generator is located in the control device, not in the drive device. In cyclic synchronous manner, it provides a target velocity to the drive device, which performs velocity control and torque control. If desired, the position control loop may be closed over the communication system. Optionally, additive velocity and torque values may be provided by the control system in order to allow a second source for velocity and/or a torque feed forward. Measured by sensors, the drive device may provide actual values for position, velocity and torque to the control device.

The cyclic synchronous velocity mode covers the following sub-functions:

- Demand value input
- Velocity capture using position sensor or velocity sensor
- Velocity control function with appropriate input and output signals
- Limitation of torque demand

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

The behavior of the control function is influenced by control parameters such as limit functions, which are externally applicable. The drive internal control function is not specified more precisely in this part of profile specification as it is highly manufacturer-specific, but the format and content of the control parameters are provided.

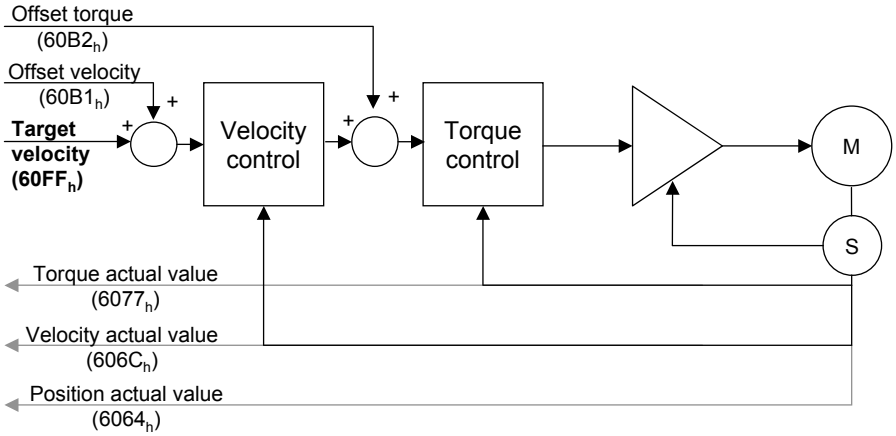


Figure 65 – Cyclic synchronous velocity mode overview

18.2 General definitions

The factors necessary for scaling have a linear relationship and therefore they are described in the factor group. The polarity is described in the factor group as well.

18.3 Functional description

Figure 66 shows the inputs and outputs of the drive control function. The input (from the control device point of view) are the target velocity and optionally, a velocity offset (to be added to the target velocity to allow two instances to set up the velocity) as well as a torque offset. Especially in cascaded control structures, where a velocity control is followed by a torque control, the output of the velocity control loop is used as an input for a further calculation in the drive device.

The drive device may support limitation of motor speed and a quick stop function for emergency reasons. The torque may be limited as well.

The interpolation time period defines the time period between two updates of the target velocity and/or additive velocity and shall be used for intercycle interpolation.

The velocity actual value is used as mandatory output to the control device. Further outputs may be the torque actual value and the velocity sensor actual value.

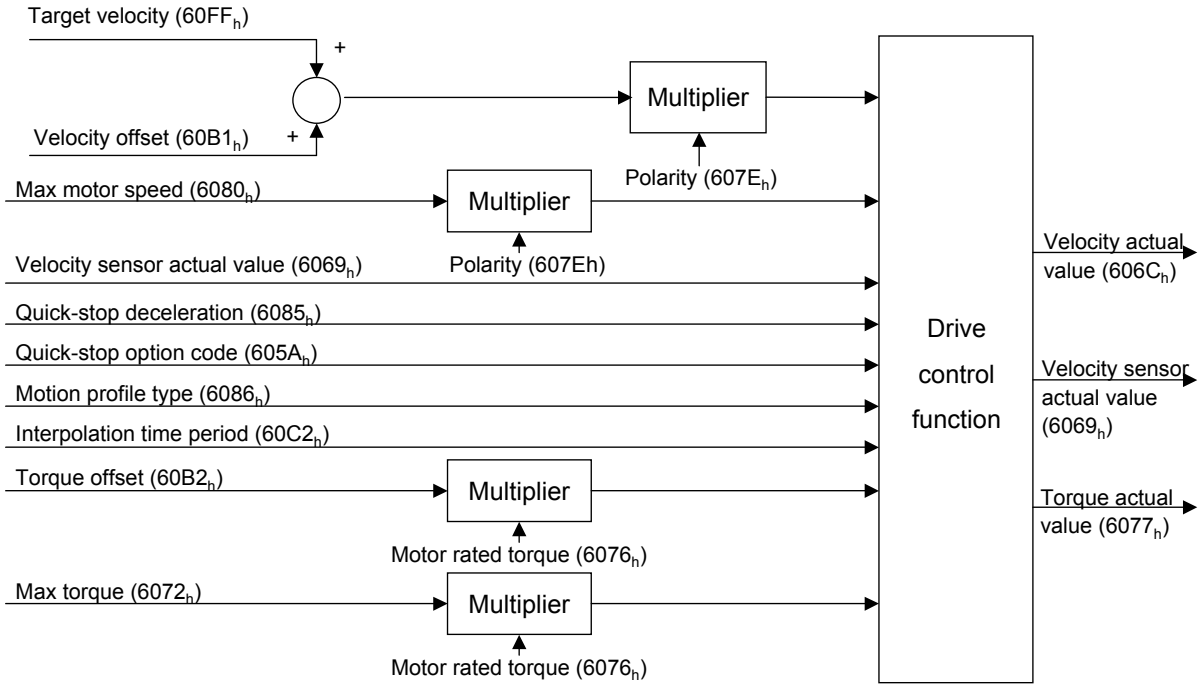


Figure 66 – Cyclic synchronous velocity control function

All values are transformed – if necessary – from user-defined units to normalised units such as increments with the functions described in Clause 9.

18.4 Use of controlword and statusword

The cyclic synchronous velocity mode uses no mode specific bits of the controlword and some bits of the statusword for mode-specific purposes. Figure 67 shows the structure of the statusword. Table 243 defines the values for bit 10, 12, and 13 of the statusword.

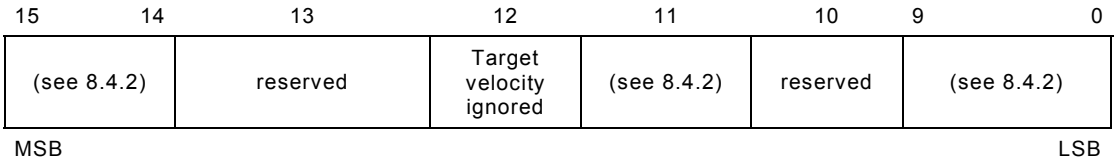


Figure 67 – Statusword for profile cyclic synchronous velocity mode

Table 243 – Definition of bit 10, bit 12, and bit 13

Bit	Value	Definition
10	0	Reserved
	1	Reserved
12	0	Target velocity ignored
	1	Target velocity shall be used as input to velocity control loop
13	0	Reserved
	1	Reserved

19 Cyclic synchronous torque mode

19.1 General information

The overall structure for this mode is shown in Figure 68. With this mode, the trajectory generator is located in the control device, not in the drive device. In cyclic synchronous manner, it provides a target torque to the drive device, which performs torque control. Optionally, an additive torque value can be provided by the control system in order to allow two instances to set up the torque. Measured by sensors, the drive device may provide actual values for position, velocity and torque to the control device.

The cyclic synchronous torque mode covers the following sub-functions:

- demand value input;
- torque capture;
- torque control function with appropriate input and output signals;
- limitation of torque demand.

The drive internal control function is not specified more precisely in this part of profile specification as it is highly manufacturer-specific, but the format and content of the control parameters are provided.

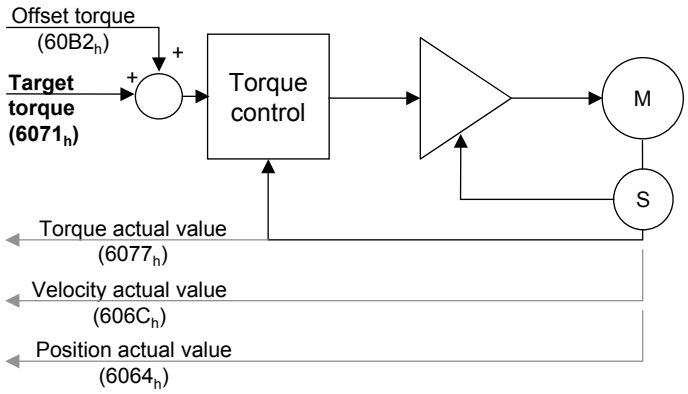


Figure 68 – Cyclic synchronous torque mode overview

19.2 General definitions

The factors necessary for scaling have a linear relationship and therefore they are described in the factor group. The polarity is described in the factor group as well.

19.3 Functional description

Figure 69 shows the inputs and outputs of the torque control function. The input (from the control function point of view) are the target torque and optionally a torque offset (to be added to the target torque to allow two instances to set up the torque).

The drive device can have features for limitation of motor speed. The torque can be limited as well.

The interpolation time period defines the time period between two updates of the target velocity and/or additive velocity and shall be used for intercycle interpolation.

The torque actual value is used as mandatory output to the control device.

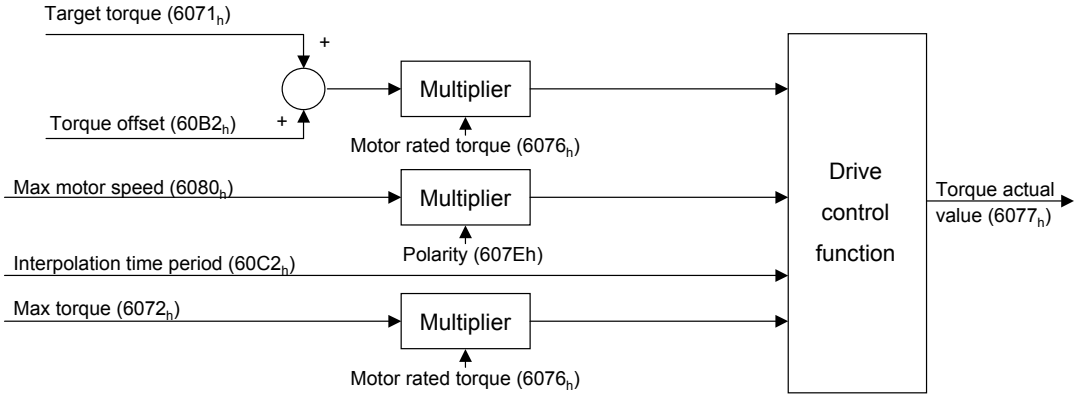


Figure 69 – Cyclic synchronous torque control function

19.4 Use of controlword and statusword

The cyclic synchronous torque mode uses no mode specific bits of the controlword and some bits of the statusword for mode-specific purposes. Figure 70 shows the structure of the statusword. Table 244 defines the values for bit 10, 12, and 13 of the statusword.

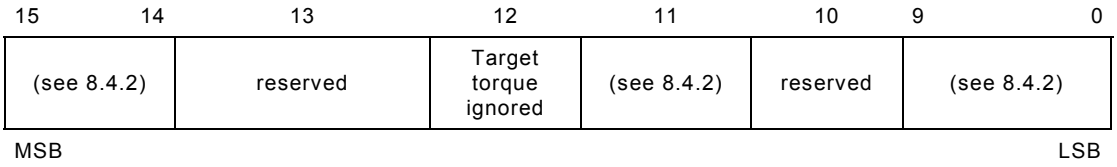


Figure 70 – Statusword for profile cyclic synchronous torque mode

Table 244 – Definition of bit 10, bit 12, and bit 13

Bit	Value	Definition
10	0	Reserved
	1	Reserved
12	0	Target torque ignored
	1	Target torque shall be used as input to torque control loop
13	0	Reserved
	1	Reserved

20 Optional application FE

20.1 General

The objects defined in this clause are used for the optional generic input/output FE.

20.2 Object 60FD$_h$: Digital inputs

This object shall provide digital inputs. Figure 71 specifies the object structure.

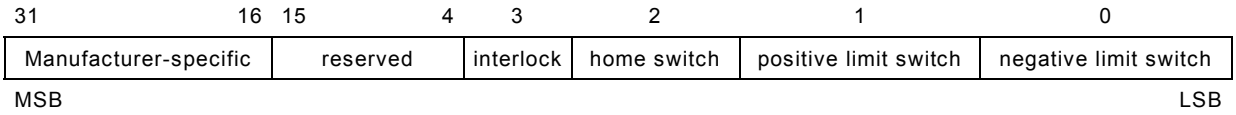


Figure 71 – Object structure

Table 245 specifies the values.

Table 245 – Value definition

Value	Definition
0_b	Switched off
1_b	Switched on

Table 246 specifies the object description, Table 247 specifies the entry description.

Table 246 – Object description

Attribute	Value
Index	$60FD_h$
Name	Digital inputs
Object code	Variable
Data type	Unsigned32
Category	Optional

Table 247 – Entry description

Attribute	Value
Sub-Index	00_h
Access	ro
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	$0000\ 0000_h$

20.3 Object $60FE_h$: Digital outputs

This object shall command simple digital outputs. Figure 72 specifies the object structure.

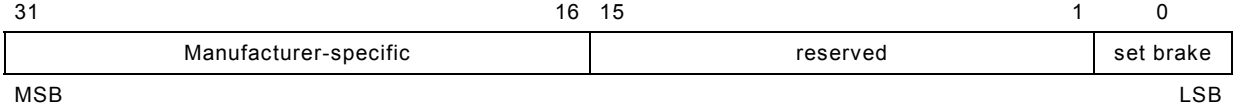


Figure 72 – Object structure

Table 248 specifies the values.

Table 248 – Value definition

Value	Definition for sub-index 01_h	Definition for sub-index 02_h
0_b	Switch off/don't set brake	Disable output
1_b	Switch on/set brake	Enable output

Table 249 specifies the object description, Table 250 specifies the entry description.

Table 249 – Object description

Attribute	Value
Index	$60FE_h$
Name	Digital output
Object code	Array
Data type	Unsigned32
Category	Optional

Table 250 – Entry description

Attribute	Value
Sub-Index	00_h
Description	Highest sub-index supported
Entry Category	Mandatory
Access	c
PDO mapping	See /CiA402-3/
Value range	~~02_h~~ 01h -02h
Default value	Manufacturer-specific
Sub-Index	01_h
Description	Physical outputs
Entry Category	Mandatory
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	$0000\ 0000_h$
Sub-Index	02_h
Description	Bit mask
Entry Category	Optional
Access	rw
PDO mapping	See /CiA402-3/
Value range	Unsigned32
Default value	$0000\ 0000_h$

CiA 402 version 3.0

Drives and motion control device profile

Part 2: Operation modes and application data

CORRIGENDUM 1

09 September 2010

Page 9, 4.2:

Add in the second sentense of the second paragraph in numerical order:

Objects 6050_h and 6051_h

Add at the end of the second paragraph the sentence:

The following object indices range shall be reserved for compatibility reasons with other standards: from 6200_h to $62FF_h$ and from 6600_h to $67EF_h$.

Page 10, *5.2:*

Replace in Table 2 *"Highest index supported"* with:

Highest sub-index supported

Replace in Table 3 *"Highest index supported"* with:

Highest sub-index supported

Page 11, *5.2*

Replace in Table 4 *"Highest index supported"* with:

Highest sub-index supported

Page 17 to 21, 7.1:

Add in Table 24 to the meaning of error code 2310_h after "Continuous over current":

(device output side)

Indent in Table 24 the meaning of the error code 3330_h like the meaning of the error code 3320_h.

Indent in Table 24 the meaning of the error code 3331_h like the meaning of the error code 3321_h.

Add in Table 24 to the meaning of error code 5100_h after "Supply":

device hardware

Add in Table 24 to the meaning of error code 5200_h after "Control":

device hardware

Add in Table 24 to the meaning of error code 7100_h after "Power":

additional modules

Indent in Table 24 the meaning of the error code 8700_h like the meaning of the error code 8600_h.

Indent in Table 24 the meaning of the error code 8800_h like the meaning of the error code 8600_h.

Add in Table 24 to the meaning of error code $8A00_h$ after “Control”:

monitoring

Indent in Table 24 the meaning of the error code $8A00_h$ like the meaning of the error code 8600_h.

Add in Table 24 to the meaning of error code $F004_h$ after “Control”:

additional functions

Add a line in the Table 24 after the error code 8612_h:

Error code 8613_h, Homing error

Replace in the last line of the Table 24 the error code *“$FF00_h$”* with:

$FF01_h$

Page 21, *8.1:*

Delete *“(not in the scope of this specification)”* from the second sentence in the second paragraph.

Page 25 to 26, 8.4.1:

Add after the third sentence:

The bits 0 to 9 shall be supported according to the mode of operation. If the related functionality is not available, an appropriate emergency message shall be generated.

Replace the fourth sentence *“The other bits may be supported.”* with:

The manufacturer-specific (ms) bits may be supported.

Add at the end of the first paragraph:

All implemented bits of the controlword are valid independent of the PDS FSA state. Starting of any movement is only possible, if a rising edge of an operation mode specific bit in the state operation enabled is received by the drive.

Page 27, 8.4.2:

Add after the third sentence:

The *oms* bits shall be supported if the mode of operation is supported. If the related functionality of the oms bits is not available, the corresponding bit shall be 0_b.

Replace the fourth sentence *"The other bits may be supported."* with:

The manufacturer-specific (ms) bits may be supported.

Add at the end of the first paragraph:

All implemented bits of the control word are valid independent of the PDS FSA state.

Add at the end of the sixth paragraph:

If the same target value is resend then bit 10 shall not alter, if bit 10 is supported.

Page 34, *8.4.10:*

Replace *"NOTE"* before Table 53 with:

NOTE 1

Add after the NOTE 1:

NOTE 2 The behavior of the drive in mode of operation 0 is manufacturer-specific

Add after the NOTE 2:

NOTE 3 If mode of operation is changed, the behavior of the drive is manufacturer-specific

Replace in Table 54 the Value of the Category with:

Conditional: mandatory, if more than one mode of operation is supported

Replace in Table 55 the Value of the Default value with:

0, if more than one mode supported; Value of the supported mode (according to Table 53), if only one mode supported

Page 35, 8.4.11:

Replace in Table 56 the Value of the Category with:

Conditional: mandatory, if more than one mode of operation is supported

Page 37, 9.2.1:

Replace in Table 61, sub-index 00_h the Value of the Default value with:

02_h

Page 38, 9.2.2:

Replace in Table 63, sub-index 00_h the Value of the Default value with:

02_h

Page 39, *9.2.3:*

Replace in Table 65, sub-index 00_h the Value of the Default value with:

02_h

Page 39, 9.2.4:

Replace the term *"output shaft"* in the first paragraph by:

driving shaft

Page 40, 9.2.4:

Replace in Table 67, sub-index 00_h the Value of the Default value with:

02_h

Page 47, *10.5.2:*

Replace in Table 76, sub-index 00_h the Value of the Default value with:

02_h

Page 48, 10.5.3:

Replace in Table 78, sub-index 00_h the Value of the Default value with:

02_h

Page 50, *10.5.6:*

Delete the text sequence after the first sentence:

allowed velocity in either direction during a profiled motion.

Delete the sentence:

It shall be converted to position increments per second using the velocity encoder factor object.

Add before the last sentence:

The speed units depend on the user-defined position units (position units per second). The calculation of the user-defined position units is done via the factor group (see clause 9).

Page 51, 10.5.8:

Delete the text sequence in the second sentence:

; it shall be converted to position increments per square second (s^2) using the normalising factors (see Clause 9).

Page 62: 11.3.10.

Replace the title *"Method 35: Homing on index pulse"* with:

Method 35: Homing on current position

Add at the end of the chapter:

This homing method is mandatory if homing mode is supported.

Page 62, 11.4:

Add before Figure 28:

Bit 12 shall provide the actual homing status. Bit 12 may be used to observe whether homing is attained without executing the homing procedure.

Page 63, 11.5.1:

Add at the end of the first paragraph (before Figure 29):

NOTE The activation of a new value of the object home offset is manufacturer-specific. It is recommended to apply the new value only while the drive is in homing mode.

Page 68, 11.5.6:

Replace in Table 119 the Definition of Bit 1, Value 1 with:

Touch probe 1 positive edge position stored

Replace in Table 119 the Definition of Bit 2, Value 1 with:

Touch probe 1 negative edge position stored

Replace in Table 119 the Definition of Bit 9, Value 1 with:

Touch probe 2 positive edge position stored

Replace in Table 119 the Definition of Bit 10, Value 1 with:

Touch probe 2 negative edge position stored

Add at the end of the Table 119:

NOTE Bit 1 and bit 2 are set to 0_b when touch probe 1 is switched off (object $60B8_h$ bit 0 is 0_b). Bit 9 and 10 are set to 0_b when touch probe 2 is switched off (object $60B8_h$ bit 8 is 0_b).

Page 68 to 69, 11.5.7

Replace in the headline and in the Table 122 *"Touch probe pos1 pos value"* with:

Touch probe 1 positive edge

Page 69, 11.5.8:

Replace in the headline and in the Table 124 *"Touch probe pos1 neg value"* with:

Touch probe 1 negative edge

Page 69 to 70, 11.5.9:

Replace in the headline and in the Table 126 *"Touch probe pos2 pos value"* with:

Touch probe 2 positive edge

Page 70, 11.5.10:

Replace in the headline and in the Table 128 *"Touch probe pos2 neg value"* with:

Touch probe 2 negative edge

Page 74, 12.3.3:

Replace in Table 134 the Value of the Category with:

Mandatory if *pp, ip, csp* is supported

Page 77, 12.3.9:

Replace the third sentence with:

The value shall be given in user-defefined position units.

Page 86, 13.4:

Replace in Figure 43 the name of bit 13 with:

following error

Replace the title of the Table 159 with:

Definition of bit 10, bit 12 and bit 13

Add in Table 159 the definition for bit 13:

Value 0: No following error. Value 1: Following error.

Page 87 to 88, 13.5.2:

Replace in Table 163 the Value of the Name with:

Interpolation data record

Replace in Table 164, sub-index 00_h the Value of the Default value with:

Manufacturer-specific

Page 92, 14.1:

Delete from the sentence before the last the sentence-part:

,but the format and maximum number of control coefficients are established.

Page 108, *16.2.1:*

Delete clause 16.2.1.

Page 109 to 110, 16.4:

Replace the title of the Figure 55 with:

Controlword for velocity mode

Replace the title of the Figure 57 with:

Statusword for velocity mode

Page 111, 16.5.3:

Replace at the end of the first paragraph the text sequence *"(velocity demand, velocity control effort, calculated velocity, measured velocity)*" with:

derived for example from velocity demand or a sensor signal.

Page 112, *16.5.4*

Replace in Figure 58 *"vl_velocity_motor_min_amount"* with:

vl velocity min amount

Replace in Figure 58 *"vl_velocity_motor_max_amount"* with:

vl velocity max amount

Page 114, 16.5.5:

Replace in Table 226 the Value of the Category with:

Optional

Page 117, 16.5.7:

Replace in Table 230 the Value of the Category with:

Optional

Page 130, 20.3

Replace in Table 250, sub-index 00_h the Value of the Value range with:

01_h to 02_h