Ethernet POWERLINK

Implementation Directive for CiA402

Drive Profile

Directive for using IEC 61800-7-201 within POWERLINK-based servo drives

Version 0.0.4

(B&R Industrial Automation GmbH)

2024

B&R Industrial Automation GmbH

POWERLINK-Office
B&R Straße 1
5142 Eggelsberg
Austria

powerlink.office@br-automation.com
www.br-automation.com/en/technologies/powerlink/

Pre. 1 Disclaimer

Use of this EPSG Standard is wholly voluntary. The EPSG disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other EPSG Standard document.

The EPSG does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. EPSG Standards documents are supplied “AS IS”.

The existence of an EPSG Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the EPSG Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Users are cautioned to check to determine that they have the latest edition of any EPSG Standard.

In publishing and making this document available, the EPSG is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the EPSG undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other EPSG Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of the EPSG, the group will initiate action to prepare appropriate responses. Since EPSG Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, the EPSG and it’s members are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of EPSG Standards are welcome from any interested party, regardless of membership affiliation with the EPSG. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be sent to the address given on the page before.

Pre 1.1 Patent notice

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. B&R shall not be responsible for identifying patents for which a license may be required by an EPSG standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Pre. 2 History

Vers.	Date	Author / Filename		Description
0.0.1	2016-04-11	Gratzl/Altmann	B&R et al.	created
0.0.2	2016-05-10	Kirchmayer	B&R	Headers adjusted
0.0.3	2018-02-01	Gratzl	B&R	Index 6094h deleted
0.0.4	2024-03-18	Kirchmayer	B&R	© B&R due to dissolution of the EPSG

Pre. 3 Content

Pre. 4 Figures

Pre. 5 Tables

1 Scope

IEC 61800-7-201 and IEC 61800-7-301 specify the CiA402 drive profile which is mapped to POWERLINK. In the IEC Standard many objects and operation modes are defined as optional therefore this document intends to define a common behavior of a POWERLINK servo drive supporting the CiA402 drive profile.

Scope of this directive
- CiA402 Servo drives and compatible frequency inverters with POWERLINK interface

NOTE Objects / functions / operation modes in bold letters throughout the text and the figures means that this feature is mandatory according to this directive.

This directive is not a substitute of the IEC 61800-7-201 specification! For the understanding and implementation of the drive profile the IEC specification is indispensable.

The directive gives explanations and defines enhancements according to the IEC specification.

1.1 IEC 61800-7

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

IEC 61800-7-1 specifies a generic interface between power drive system(s) (PDS) and the application control program in a controller. The generic PDS interface is not specific to any particular communication network technology. Annexes of IEC 61800-7-1 specify the mapping of the different drive profiles onto the generic PDS interface.

IEC 61800-7-201 specifies the CiA402 DriveProfile for power drive systems (PDS). This profile of the PDS is not specific to any particular communication network technology.

IEC 61800-7-301 specifies the mapping of different profiles to different network technologies.

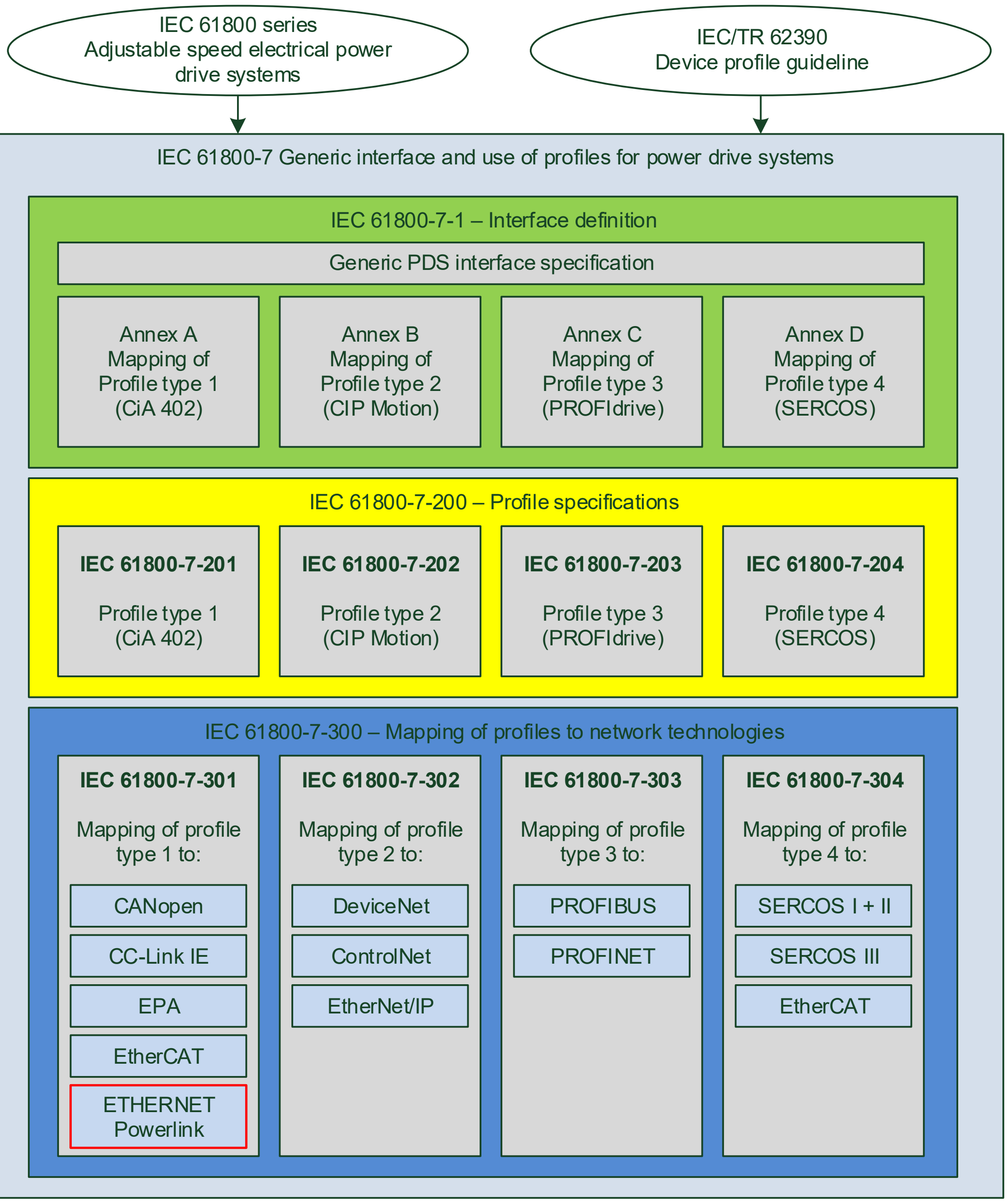

Figure 1: Structure of the IEC 61800-7

2 References

[1] IEC 61800-7-201: Adjustable speed electrical power drive systems Part 7-2 Generic interface and use of profiles for power drive systems Profile specification Type 1

[2] IEC 61800-7-301: Adjustable speed electrical power drive systems Part 7-2 Generic interface and use of profiles for power drive systems mapping of profiles to network technologies

3 Terms and Definitions

(M) Mandatory
the object / feature shall be supported according to IEC 61800-7 or according to this directive.

(C) Conditional
the object / feature parameter is conditional upon other parameters/features.

I.e. it shall be supported if (the) other parameter(s)/feature(s) is(are) supported.

(R) Recommended
the object / feature is highly recommended according to this directive.
It is defined for a better quality of the corresponding function.

(O) Optional
the object / feature might be supported.
It is not recommended according to this directive.

Low-level power
electrical power supply for the control section of the drive, e.g. 24V.

High-level power
main electric power supply of the drive, e.g. 230V or 380V.

3.1 Abbreviated terms

CiA	CAN in Automation
csp	Cyclic Synchronous Profile mode
cst	Cyclic Synchronous Torque mode
csv	Cyclic Synchronous Velocity mode
CSM	Communication State Machine
FSA	Finite State Automat
hm	Homing mode
ip	Interpolated Position mode
ms	Manufacturer Specific
PDS	Power Drive System
pp	Profile Position mode
pv	Profile Velocity mode
Res, r	Reserved
ro	Read-Only
rw	Read-Write
SM	State Machine
vl	Velocity mode
wo	Write-Only

4 Mapping to POWERLINK

4.1 Mapping of communication objects

Communication objects for real-time data transmission shall be mapped to Process Data Object message (PDOs). Communication objects for configuration data transmission shall be mapped to Service Data Object message(s) (SDO).
System time distribution and synchronization is handled implicitly by the POWERLINK protocol without the need for explicit messages. [61800-7-301]

4.1.1 Process Data Object (PDO)

The real-time data transfer is performed by means of Process Data Objects (PDO).

PDO communication in POWERLINK is always performed isochronously by PReq and/or PRes frames. PReq frames are sent as unicast from the Managing Node to Controlled Nodes. Each frame contains real-time output data for a single node. PRes frames are broadcast to the whole network and may therefore contain data directed to several nodes. This allows direct cross-communication following the producer/consumer principle.

The transmission type of PDO is continuous. There is no “on event” or “on change” transmission type provided.

From the device’s view, there are two types of PDO usage: data transmission and data reception. Transmit PDOs (TPDOs) and Receive PDOs (RPDOs) shall be distinguished.

The mapping of the process data is not especially fixed by this directive.
Depending on the supported modes some objects must be mapped to TPDO/RPDO and some are optional (O) to be mapped to PDO.
e.g.: 8.2.4 Object list

4.1.2 Service Data Object (SDO)

The Service Data Object protocol allows asynchronous (i.e. non-real-time) access to the device. The SDO protocol follows the master/slave principle and allows access to the whole object dictionary via index and sub-index addressing.

SDO is used to manipulate the objects on the drive.

NOTE: Access (rw, ro, wo) via SDO shall be supported for each object defined on the drive!

5 Device profile

5.1 Object 1000_h: DeviceType_U32 (M)

The device type object 1000h shall define the device type, the device's functionality, and the mapping variant. [61800-7-301]

Device	Additional information									Device profile number
	Mode bits	Type								
	31…24	23	22	21	20	19	18	17	16	15…0
Frequency converter	manufacturer specific	0	x	0	0	0	0	0	1	0192_h (402_d)
Servo drive	manufacturer specific	0	x	0	0	0	0	1	0	0192_h (402_d)

Figure 2: Object 1000_h Device Type

The Device profile number (Bit 0…15) shows, if the device corresponds with the CiA402 standard.

6 Error codes and error behavior

Emergency messages are triggered by internal errors and severe warnings detected within the drive device.

Error numbers are:

- defined in IEC 61800-7-201
- 16 Bit values
- numbers from XX00h to XX7Fh
- numbers from XX80h to XXFFh (manufacturer-specific)

6.1 Object 603F$_h$: ErrorCode_U16 (O)

This object shall provide the error code of the last error which occurred in the drive device.
In CANopen this object provides the same information as the lower 16-bit of sub-index 01h of the pre-defined error field (1003h).

6.2 Error behavior

The following error option codes are supported:

Index	Name	Category	Access	Object	Type	PDO - mapping
6007$_h$	AbortConnectionOptionCode_I16	O	RW	VAR	INTEGER16	O
605A$_h$	QuickStopOptionCode_I16	O	RW	VAR	INTEGER16	O
605B$_h$	ShutdownOptionCode_16	O	RW	VAR	INTEGER16	O
605C$_h$	DisableOperationOptionCode_I16	O	RW	VAR	INTEGER16	O
605D$_h$	HaltOptionCode_I16	O	RW	VAR	INTEGER16	O
605E$_h$	FaultReactionOptionCode_I16	O	RW	VAR	INTEGER16	O

Table 1 - Supported Error option codes

The communication state machine (CSM) is independent of the drive state machine (FSA SM).
Object 0x6007 can be used to define the behavior in case the slave CSM leaves the OP state.
A state change of the FSA SM does not affect the CSM.

7 Controlling the power drive system

7.1 State machine

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.

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.

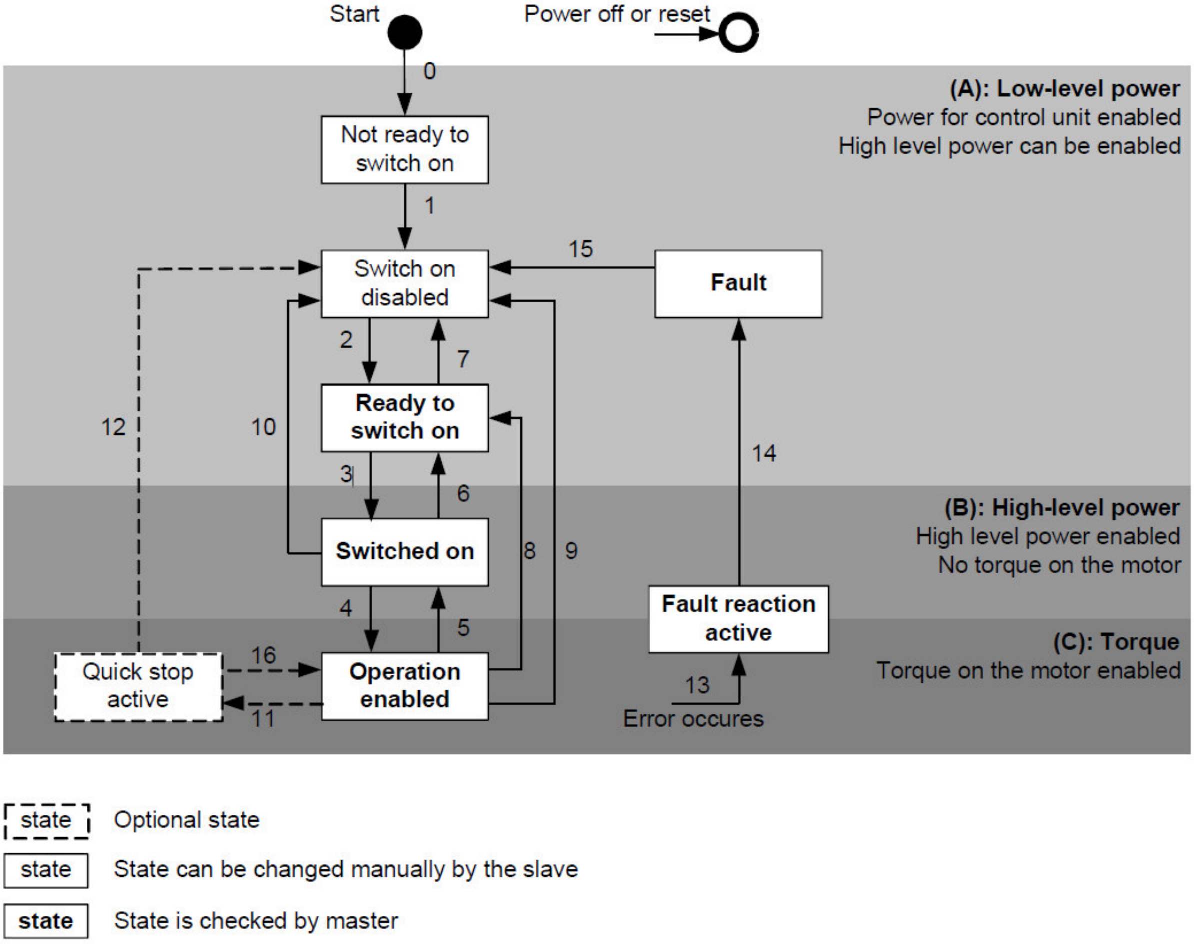

Figure 3: State Machine

Transition	Event(s)	Action(s)
0	Automatic transition after power-on or reset application	Drive device self-test and/or self-initialization 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 IEC 61800-7-301)	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.
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_U16 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.		

Table 2 - Transition events and actions

The state *Quick stop active* is optional. Usually the control device will perform the quick stop function.

The state *Not ready to switch on* is performed automatically by the drive.

The state *Switch on disabled* can be passed through automatically by local signals of the drive. The control device has to compare during start-up which state is reached by the drive. Then the control device has to perform the transition *2* only if necessary.

The states in bold letters are stable states in the drive and will be checked by the control device. The transitions *3* and *4* can only be requested by the control device.

If the control device sets Bits 0, 1 and 3 simultaneously in the ControlWord_U16 the drive can pass from *Ready to Switch on* via *Switched on* to *Operation enabled* with one control device command.

In area (A) the Low-level power is enabled. The High-level power may be enabled, e.g. to supply the low-level power.

In area (B) the High-level power shall be switched on but there is no torque on the motor. Transition *3* shall be refused by the drive, if no high-level power is enabled. Target and set-point values shall be ignored.

In area (C) the drive is ready to operate and the torque is switched on to the motor. Target and setpoint values shall be processed.

During transition *4* the torque can be set up and if necessary a brake can be opened. The StatusWord_U16 displays the state *Switched on* until the drive is ready to follow the target and set-point values.

Accordingly in transition 5 the drive can actively be decelerated and the brake can be closed. The StatusWord_U16 displays the state Operation enabled as long as torque on the motor is enabled.

The servo drive shall support the following 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
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

Table 3 - FSA states and supported functions

7.2 Object 6040h: ControlWord_U16 (M)

This object shall indicate the received command controlling the PDS FSA.
The following Bits in the ControlWord_U16 shall be supported.

15 - 11	10	9	8	7	6	5	4	3	2	1	0
Manu-facturer specific	Res.	**Operation mode specific**	**Halt**	**Fault reset**	**Operation mode specific**			**Enable Operation**	**Quick stop**	**Enable voltage**	**Switch on**

Figure 4: ControlWord_U16 value definition

Bit 2 Quick stop shall be set (= 1) by the control device to deactivate an optional quick stop. This bit shall be ignored by a drive that does not support the quick stop state.

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.						

Table 4 - Command coding

7.3 Object 6041h: StatusWord_U16 (M)

This object shall provide the status of the PDS FSA.
The following Bits in the StatusWord_U16 shall be supported:

15,14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	operation mode specific		internal limit active	target reached	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 5: StatusWord_U16 value definition

Bit 5 Quick Stop shall be set by the drive if the Quick Stop state is not supported or the Quick stop function is not active.
Bit 9 Remote shall be set if the drive is controlled by the control device.
Bit 11 Internal limit active is set if internal limits are exceeded so that the target and set-point values can't be reached. E.g. for hardware position switches, current limiter or thermal overload.

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

Table 5 - State coding

8 Modes of operation

8.1 General

8.1.1 Object 6060h: ModesOfOperation_I8 (M)

This object shall indicate the requested operation mode.
The following operation modes for the servo drives are recommended.

Mode of operation	Abbr.	Code
Manufacturer specific mode	-	-128..-1
no mode change/no mode assigned	-	0
Profile position mode	pp	1
Velocity mode	vl	2
Profile velocity mode	pv	3
Torque profile mode	tq	4
reserved	-	5
Homing mode	hm	6
Interpolated position mode	ip	7
Cyclic synchronous position mode	csp	8
Cyclic synchronous velocity mode	csv	9
Cyclic synchronous torque mode	cst	10
Cyclic synchronous torque mode with commutation angle	cstca	11
reserved	-	12 - 127

Table 6 - Operation Modes

8.1.2 Object 6061h: ModesOfOperationDisplay_I8 (M)

This object shall provide the actual operation mode.
Table 6 specifies the value definition.

8.1.3 Object 6502h: SupportedDriveModes_U32 (O)

This object shall provide information on the supported drive modes.

31 … 16	15 … 11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific	reserved	cstca	cst	csv	csp	ip	hm	r	tq	pv	vl	pp

Figure 6: Supported drive modes value definition

Depending on which Bits are set, the associated modes are supported.

8.1.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
6060_h	ModesOfOperation_I8	M	RW	VAR	INTEGER8	RPDO
6061_h	ModesOfOperationDisplay_I8	M	RO	VAR	INTEGER8	TPDO
$605B_h$	SupportedDriveModes_U32	O	RO	VAR	UNSIGNED32	O

Table 7 - Object list ModesOfOperation_I8

8.1.5 Switching between Operation modes

The operation mode can be switched through the object 6060h ModesOfOperation_I8. This should be done cyclically via PDO communication.

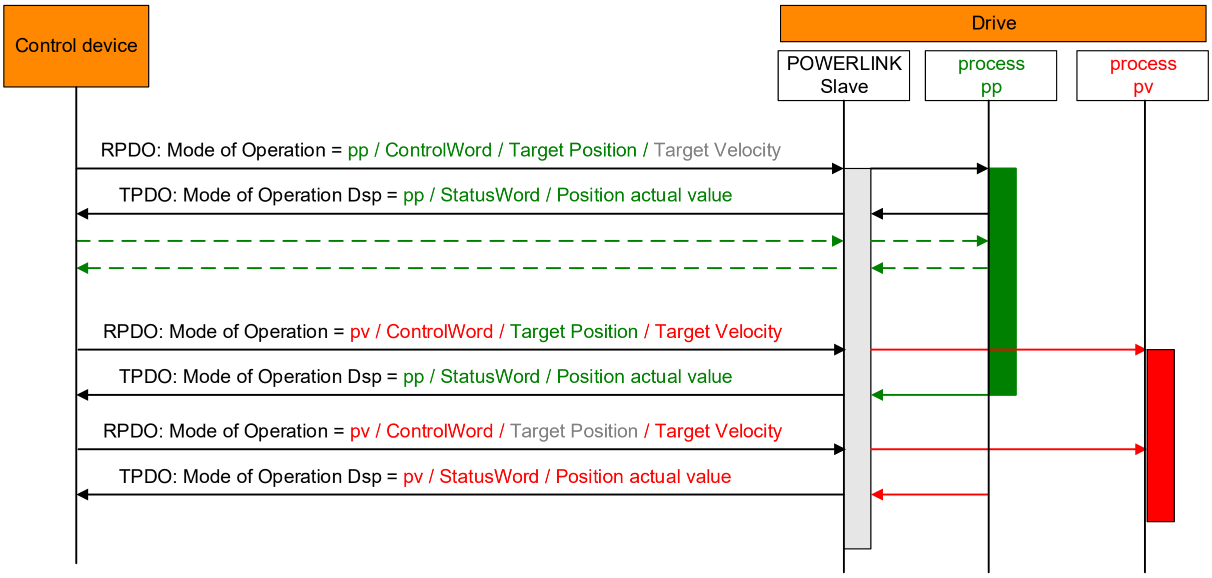

Figure 7: Dynamic change of operation mode

8.2 Profile position mode

8.2.1 General information

The overall structure for this mode is shown in Figure 8. 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. These two function blocks are optionally controlled by individual parameter sets.

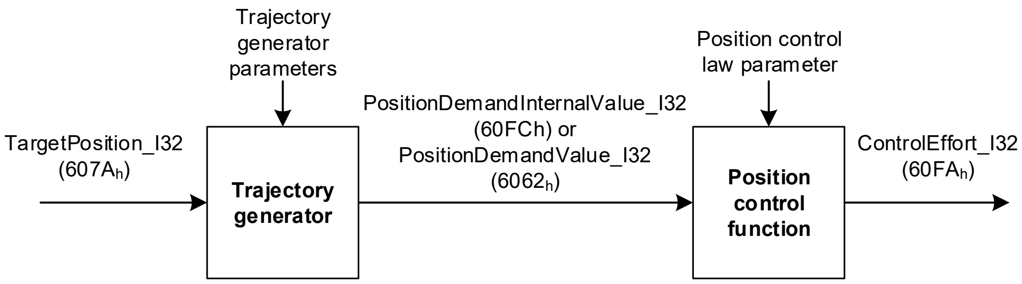

Figure 8: Trajectory generator and position control function

At the input to the trajectory generator, parameters may have optional limits applied before being normalized 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 9 defines the detailed structure of the trajectory generator.

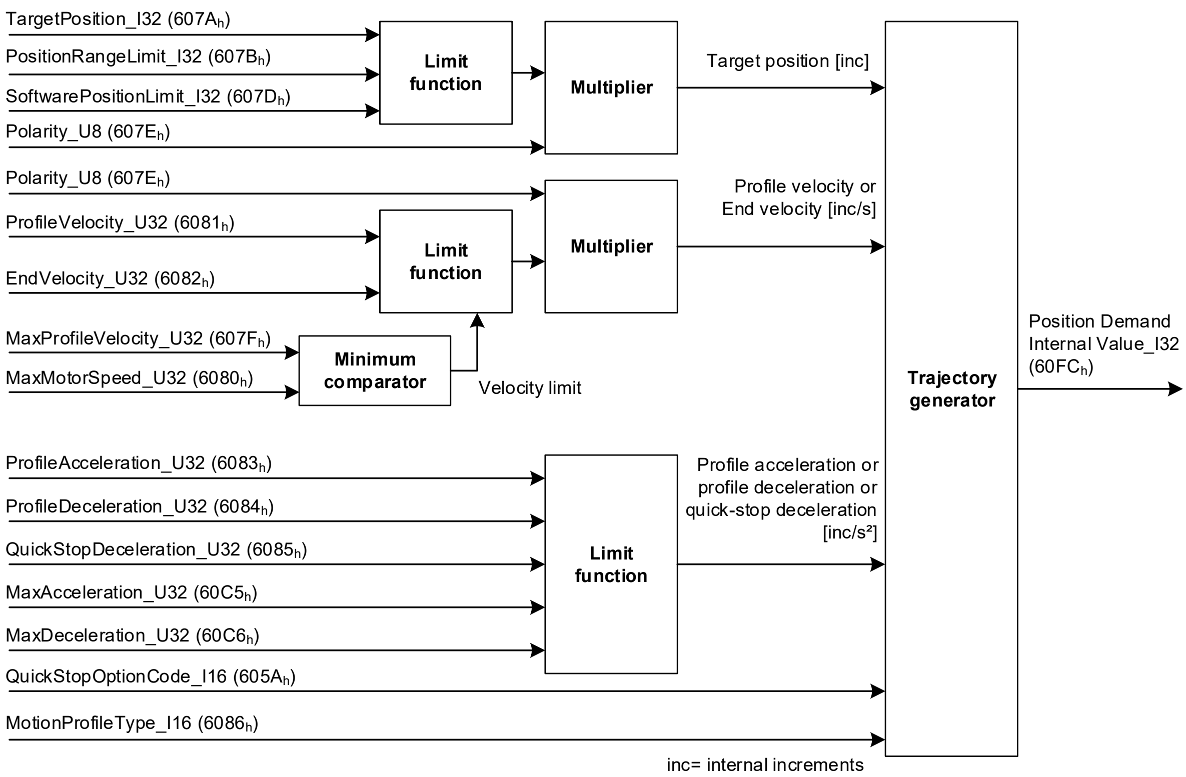

Figure 9: Trajectory generator for profile position mode

8.2.2 Functional description

8.2.2.1 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_U16 as well as the *set-point acknowledge* bit in the StatusWord_U16.
If the *change set immediately* bit of the ControlWord_U16 is set to 1, a single set-point is expected by the drive device.
If the *change set immediately* bit of the ControlWord_U16 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_U16. The drive device sets the set-point acknowledge bit in the StatusWord_U16 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 10.

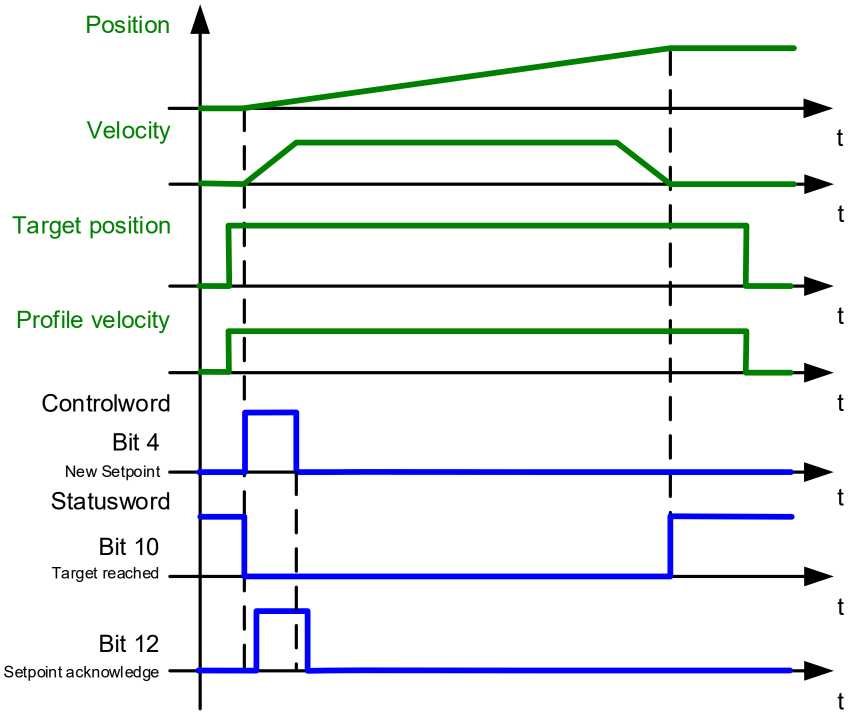

Figure 10: Set-point example

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_U16 is 1) and *set of setpoints* (*change set immediately* bit of ControlWord_U16 is 0).

8.2.2.2 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_U16, the new set-point shall be processed immediately. The handshaking procedure shown in Figure 11 is used for the single set-point method.

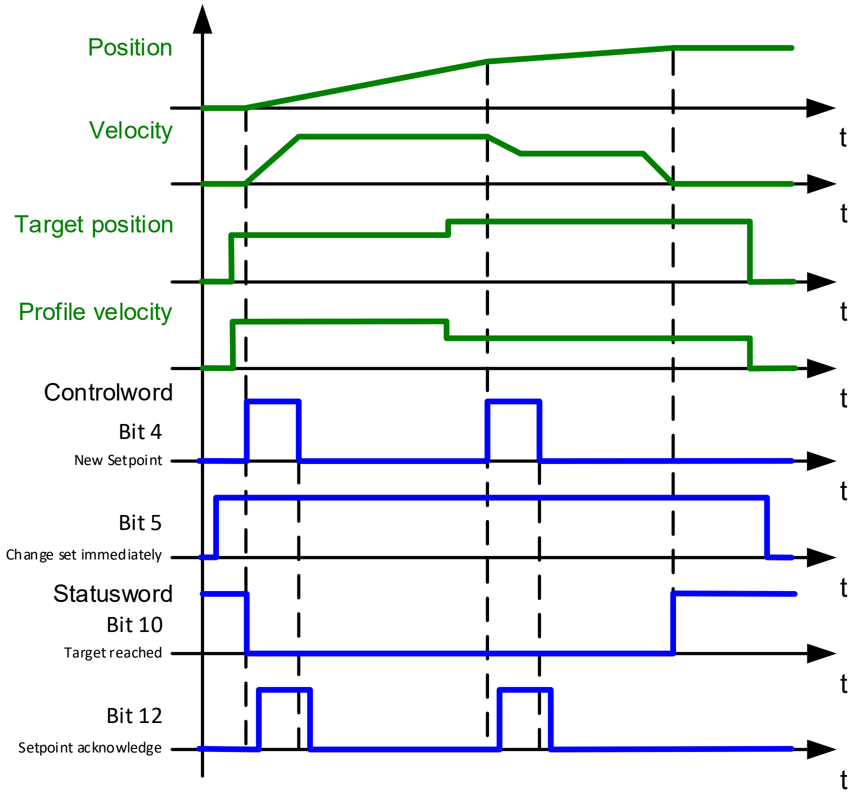

Figure 11: Handshaking procedure for the single set-point method

8.2.2.3 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_U16, the new set-point shall be processed only after the previous has been reached. The handshaking procedure shown in Figure 12 is used for the set of set-points method. The right side of the graph 'Velocity' shows the actual speed if the *change of set point* bit (bit 9) is set to 1.

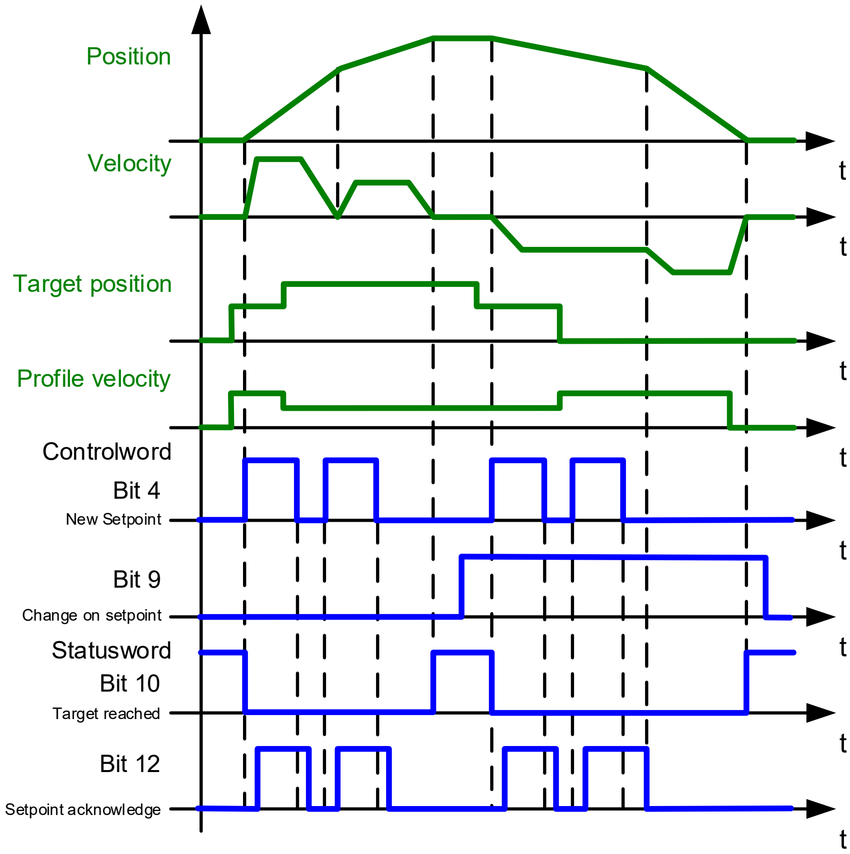

Figure 12: Handshaking procedure for the set of set-points method

If a drive device supports set of set-points, a minimum of two set-points are available, a setpoint that is currently been processed and a buffered set-point. The set-points are handled as shown in Figure 13.

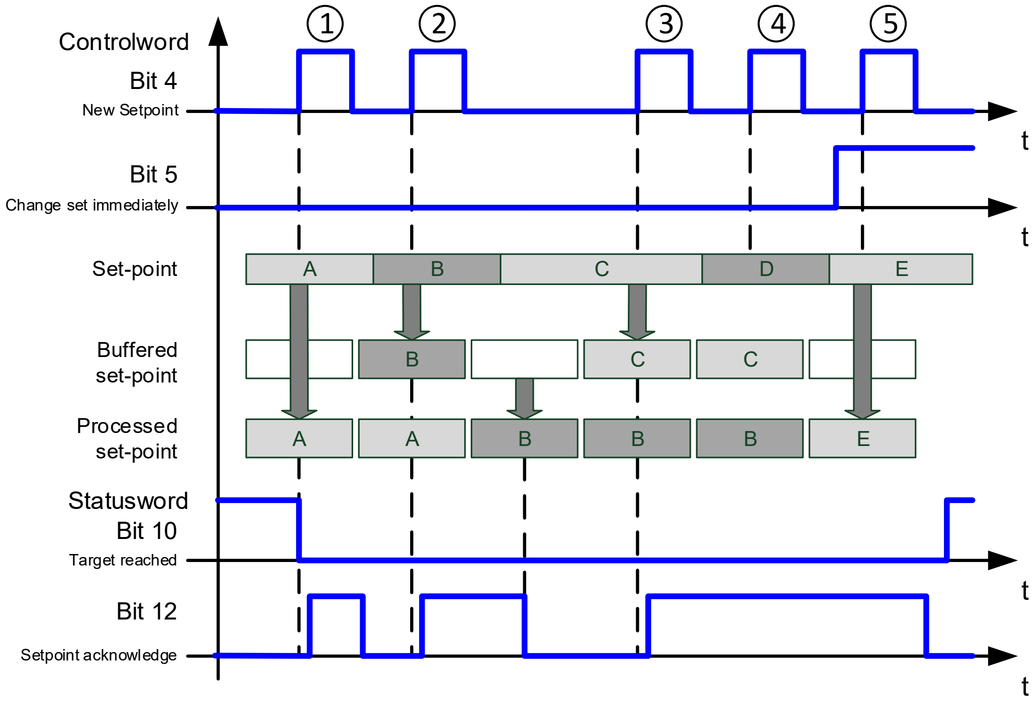

Figure 13: Set-point handling for two set-points

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.

8.2.2.4 Usage of halt-bit in conjunction with new set-point bit

The *halt-bit* can be used to halt, set a new set-point and in combination with the *new set-point bit* to interrupt an actual movement which is described in this section.

If a drive is processing a set-point the *halt-bit* (bit 8 of the ControlWord_U16) can be used to stop the motor and keep it in position control. After releasing the *halt-bit* the processing of the actual set-point is continued. Figure 21 describes a mechanism how to erase the processed set-point and set the actual position as a new set-point for that the drive remains at the current position in position control even after releasing the *halt-bit*.

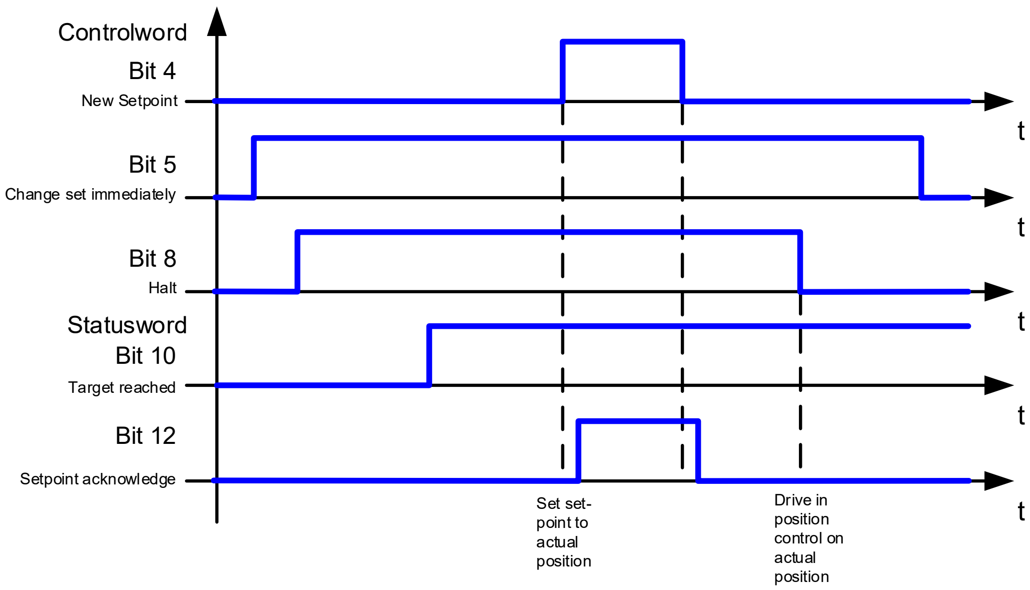

Figure 14: Erase set-point

The actual drive position shall be captured after a complete stand still of the motor. Therefore bit 8 (*halt*) of the ControlWord_U16 (6040h) and bit 10 (*target reached*) of the StatusWord_U16 (6041h) have to be equal to 1. Additional information on this bit combination can be found in Table 10.

To define the slowdown ramp which is applied during the halt command object 605Dh (HaltOptionCode_I16) can be used.

8.2.3 Use of ControlWord_U16 and StatusWord_U16

The profile position mode uses some bits of the ControlWord_U16 and the StatusWord_U16 for mode-specific purposes. Figure 15 shows the structure of the ControlWord_U16. 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 8 shall be used. Table 9 defines the values for bit 6 and 8 of the ControlWord_U16.

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

15-11	10	9	8	7	6	5	4	3	2	1	0
Manu-facturer specific	Res.	**Change on setpoint**	Halt	Fault reset	**abs/rel**	**Change set immediately**	**New Setpoint**	Enable Operation	Quick stop	Enable voltage	Switch on

Figure 15: ControlWord_U16 for profile position (pp) mode

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

Table 8 - Definition of bit 4, bit 5 and bit 9 for profile position mode

Bit	Value	Definition
6	0	Target position shall be an absolute value
	1	Target position shall be a relative value (depending on object 60F2h)
8	0	Positioning shall be executed or continued
	1	Axis shall be stopped accordingly to halt option code (605Dh)

Table 9 - Definition of bit 6 and bit 8 for profile position mode

Figure 16 shows the structure of the StatusWord_U16.
Table 10 defines the values for bit 10, bit 12, and bit 13.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**following error**	**Set-point acknow-ledge**	internal limit active	**target reached**	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 16: StatusWord_U16 for profile position (pp) mode

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

Table 10 - Definition of bit 10, bit 12 and bit 13 for profile position mode

8.2.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$607A_h$	TargetPosition_I32	M	RW	VAR	INTEGER32	RPDO
6081_h	ProfileVelocity_U32	M	RW	VAR	UNSIGNED32	RPDO
6083_h	ProfileAcceleration_U32	M	RW	VAR	UNSIGNED32	RPDO
6084_h	ProfileDeceleration_U32	M	RW	VAR	UNSIGNED32	RPDO
6064_h	PositionActualValue_I32	M	RO	VAR	INTEGER32	TPDO
$607D_h$	SoftwarePositionLimit_I32	R	RW	ARRAY	INTEGER32	O
6085_h	QuickStopDeceleration_U32	R	RW	VAR	UNSIGNED32	O

Table 11 - Object list profile position mode

8.3 Profile Velocity mode

8.3.1 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

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

8.3.2 Functional description

Figure 17 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_U16 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_U16. Below this threshold, the bit shall be set to 1 and shall indicate that the axis is stationary.

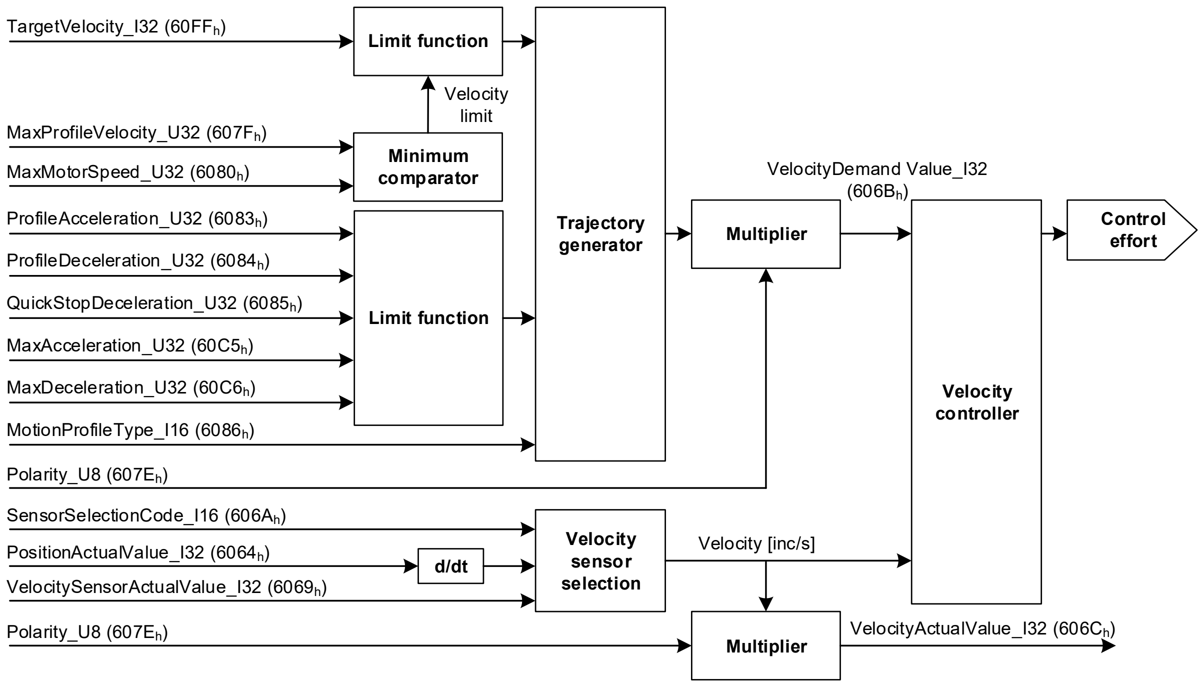

Figure 17: Profile velocity mode

8.3.3 Use of ControlWord_U16 and StatusWord_U16

The profile velocity mode uses no mode specific bits of the ControlWord_U16.
Table 12 defines the values for bit 8 of the ControlWord_U16.

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

Table 12 - Definition of bit 8 for profile velocity mode

Figure 18 shows the structure of the StatusWord_U16.
Table 13 defines the values for bit 10, 12, and 13 of the StatusWord_U16.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**Max slippage error**	**Speed**	internal limit active	**target reached**	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 18: StatusWord_U16 for profile velocity mode

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

Table 13 - Definition of bit 10, bit 12 and bit 13 for profile velocity mode

8.3.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
0x60FF	TargetVelocity_I32	M	RW	VAR	INTEGER32	RPDO
0x6083	ProfileAcceleration_U32	M	RW	VAR	UNSIGNED32	O
0x6084	ProfileDeceleration_U32	M	RW	VAR	UNSIGNED32	O
0x606C	VelocityActualValue_I32	M	RO	VAR	INTEGER32	TPDO
0x6085	QuickStopDeceleration_U32	R	RW	VAR	UNSIGNED32	O

Table 14 - Object list profile velocity mode

8.4 Interpolated position mode

8.4.1 General information

The interpolated position mode is used to control 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.
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.

8.4.2 Functional description

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. Figure 19 shows the defined structure of the profile velocity mode.

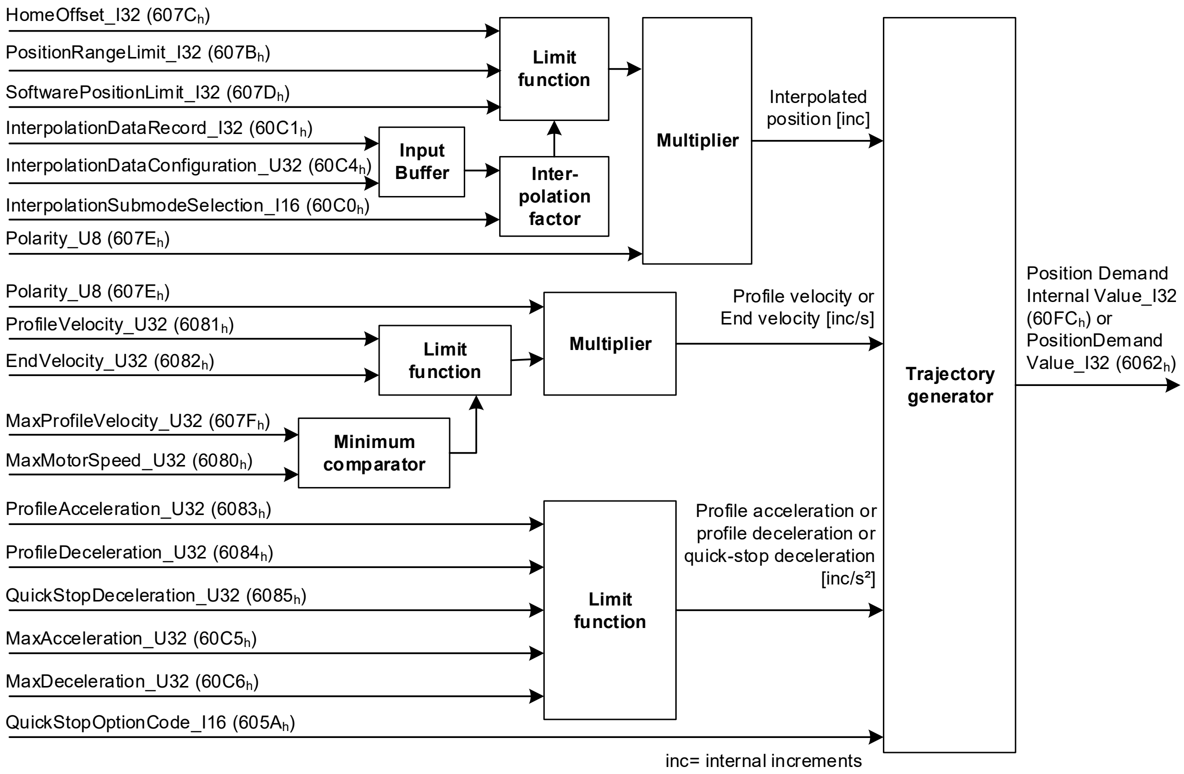

Figure 19: Interpolation controller

8.4.3 Use of ControlWord_U16 and StatusWord_U16

The interpolation position mode uses some bits of the ControlWord_U16 and the StatusWord_U16 for mode-specific purposes. Figure 20: Controlword for interpolation position mode shows the structure of the ControlWord_U16. If no positioning is in progress, the rising edge of bit 4 shall start the positioning of the axis. Table 15 defines the values for bit 4 and 8 of the ControlWord_U16.

15-11	10	9	8	7	6	5	4	3	2	1	0
Manu-facturer specific	Res.	Reserved	Halt	Fault reset	Reserved	Reserved	**Enable Inter-polation**	Enable Operation	Quick stop	Enable voltage	Switch on

Figure 20: Controlword for interpolation position mode

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 (605Dh) and Bit 12 in the statusword shall be set to 0

Table 15 - Defined values for bit 4 and bit 8 in interpolated position mode

Figure 21: Statusword for interpolated position mode shows the structure of the StatusWord_U16. Table 16 - Definition of bit 10, bit 12 and bit 13 in interpolated position mode defines the values for bit 10, 12, and 13 of the StatusWord_U16.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**following error**	**ip mode active**	internal limit active	**target reached**	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 21: Statusword for interpolated position mode

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	Interpolation inactive
	1	Interpolation active
13	0	No following error
	1	Following error

Table 16 - Definition of bit 10, bit 12 and bit 13 in interpolated position mode

8.4.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$60C0_h$	InterpolationSubModeSelect_I16	R	RW	VAR	INTEGER16	O
$60C1_h$	InterpolationDataRecord_I32	M	RW	ARRAY	INTEGER32	RPDO
$60C2_h$	InterpolationTimePeriod_U8	M	RW	ARRAY	UNSIGNED8	O
$60C4_h$	InterpolationDataConfiguration_U32	R	RW	ARRAY	UNSIGNED32	O

Table 17 - Object list interpolated position mode

8.5 Cyclic synchronous position mode

If this mode is supported the cyclic synchronous velocity mode has to be supported too.

8.5.1 General information

The overall structure for this mode is shown in Figure 22. 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 feed forward.
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.

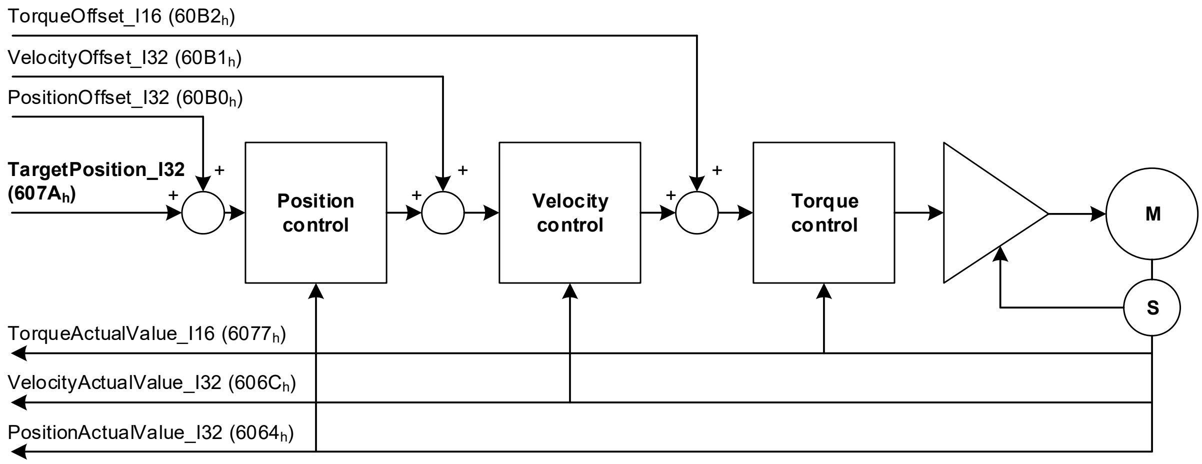

Figure 22: Cyclic synchronous position mode overview

8.5.2 Functional description

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 feed forward 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.

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_U16 to 1.

8.5.3 Use of ControlWord_U16 and StatusWord_U16

The cyclic synchronous position mode uses no mode specific bits of the ControlWord_U16.

Figure 23 shows the structure of the StatusWord_U16.
Table 18 defines the values for bit 12 and 13 of the StatusWord_U16.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**following error**	**Drive follows the command**	internal limit active	reserved	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 23: StatusWord_U16 for cyclic synchronous position mode

Bit	Value	Definition
12	0	Drive does not follow the command value – Target position ignored
	1	Drive follows the command value – Target position used as input to position control loop
13	0	No following error
	1	Following error

Table 18 - Definition of bit 12 and bit 13 for cyclic synchronous position mode

8.5.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$607A_h$	TargetPosition_I32	M	RW	VAR	INTEGER32	RPDO
$60B0_h$	PositionOffset_I32	R	RW	VAR	INTEGER32	O
$60B1_h$	VelocityOffset_I32	R	RW	VAR	INTEGER32	O
$60B2_h$	TorqueOffset_I16	R	RW	VAR	INTEGER16	O
6064_h	PositionActualValue_I32	M	RO	VAR	INTEGER32	TPDO
$606C_h$	VelocityActualValue_I32	M	RO	VAR	INTEGER32	TPDO
6077_h	TorqueActualValue_I16	R	RO	VAR	INTEGER16	O

Table 19 - Object list cyclic synchronous position mode

8.6 Cyclic synchronous velocity mode (C)

This mode has to be supported if cyclic synchronous position mode is supported.

8.6.1 General information

The overall structure for this mode is shown in Figure 24. 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

The behavior of the control function is influenced by control parameters like limit functions, which are externally applicable.

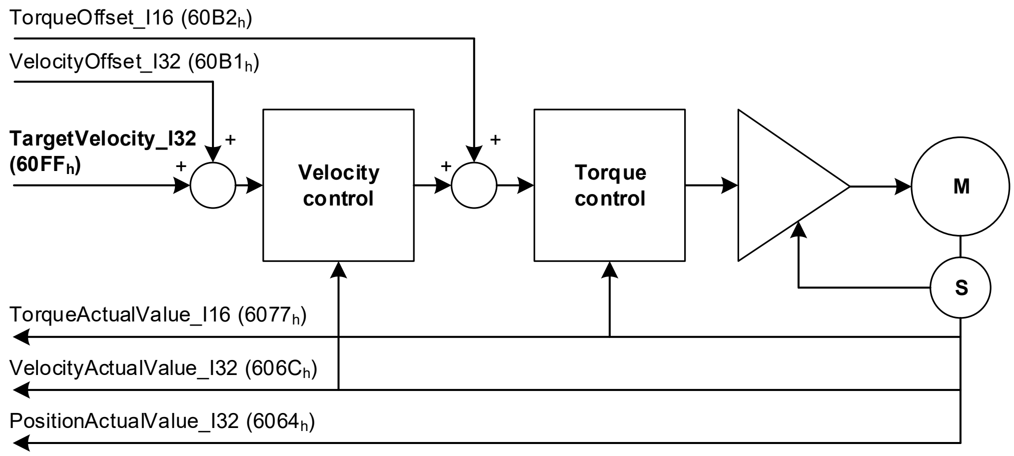

Figure 24: Cyclic synchronous velocity mode overview

8.6.2 Functional description

The inputs (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.

8.6.3 Use of ControlWord_U16 and StatusWord_U16

The cyclic synchronous velocity mode uses no mode specific bits of the ControlWord_U16.

Figure 25 shows the structure of the StatusWord_U16.
Table 20 defines the values for bit 12 of the StatusWord_U16.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**following error**	**Drive follows the command**	internal limit active	reserved	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 25: StatusWord_U16 for cyclic synchronous velocity mode

Bit	Value	Definition
12	0	Drive does not follow the command value – Target velocity ignored
	1	Drive follows the command value – Target velocity used as input to velocity control loop

Table 20 - Definition of bit 12 for cyclic synchronous velocity mode

8.6.4 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$60FF_h$	TargetVelocity_I32	M	RW	VAR	INTEGER32	RPDO
$60B1_h$	VelocityOffset_I32	R	RW	VAR	INTEGER32	O
$60B2_h$	TorqueOffset_I16	R	RW	VAR	INTEGER16	O
6064_h	PositionActualValue_I32	M	RO	VAR	INTEGER32	TPDO
$606C_h$	VelocityActualValue_I32	M	RO	VAR	INTEGER32	TPDO
6077_h	TorqueActualValue_I16	R	RO	VAR	INTEGER16	O

Table 21 - Object list cyclic synchronous velocity mode

9 Homing

9.1 General

This clause describes the method by which a drive seeks the home position. 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 use the index (zero) pulse train from an incremental encoder.

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.

Method	Description
-128 to -1	Manufacturer-specific
0	No homing method assigned
1	Homing on negative limit switch and index pulse
2	Homing on positive limit switch and index pulse
3, 4	Homing on positive home switch and index pulse
5, 6	Homing on negative home switch and index pulse
7…14	Homing on home switch and index pulse
15, 16	Reserved
17…30	Homing without index pulse
31, 32	Reserved
33, 34	Homing on index pulse
35	Homing on current position
36	reserved
37	Homing on current position
38 to 127	reserved

Table 22 - Homing Methods

9.2 Calculation of position actual value by homing process

The current position is the unprocessed position sensor information (incremental, single or multi turn sensor) before homing.

For a single turn sensor the single turn information represents the position sensor information. For a multi turn sensor the multi turn information represents the position sensor information. For an incremental sensor (e.g. TTL encoder) the position sensor information is zero after initialization.

The Home Offset is configured with the machine specific value during commissioning.

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.

Zero position = home position + home offset

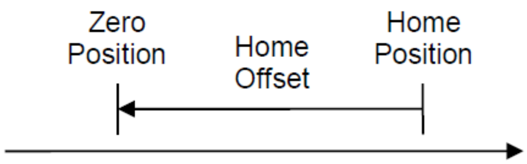

Figure 26: Home offset definition

The PositionActualValue_I32 (0x6064) is the current software position in the drive. It is based on the unprocessed position sensor information and the Home Offset (after Homing process).

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.

At the Home position (i.e. after the homing process) the PositionActualValue_I32 (0x6064) is calculated according to the used Homing method.

For example (using Methode 37):

- Before homing:
    - PositionActualValue_I32 0x6064 = 245
    - HomeOffset_I32 0x607C = 100
- After homing
    - PositionActualValue_I32 0x6064 = 100

9.3 Homing object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$607C_h$	HomeOffset_I32	M	RW	VAR	INTEGER32	O
6098_h	HomingMethod_I8	M	RW	VAR	INTEGER8	O
6099_h	HomingSpeed_U32	M	RW	ARRAY	UNSIGNED32	O
$609A_h$	HomingAcceleration_U32	M	RW	VAR	UNSIGNED32	O
$60E3_h$	SupportedHomingMethods_I8	R	RO	VAR	INTEGER8	O

Table 23 - Homing Object list

In the ControlWord_U16 the Bit 4 is used.

15-11	10	9	8	7	6	5	4	3	2	1	0
Manu-facturer specific	Res.	Reserved	Halt	Fault reset	Reserved	Reserved	**Homing operation start**	Enable Operation	Quick stop	Enable voltage	Switch on

Figure 27: ControlWord_U16 for homing

In the StatusWord_U16 the bits 12 and 13 are used.

15, 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms	**homing error**	**homing attained**	internal limit active	target reached	remote	ms	warning	switch on disabled	quick stop	voltage enabled	fault	operation enabled	switched on	ready to switch on

Figure 28: StatusWord_U16 for homing

9.4 Object 60E3h: Supported Homing Methods

This object defines the supported homing methods of the drive.

9.5 Homing Methods

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

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

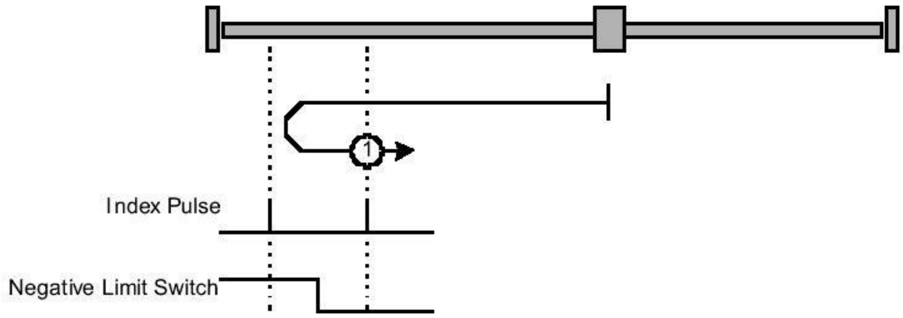

Figure 29: Homing on the negative limit switch and index pulse

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

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

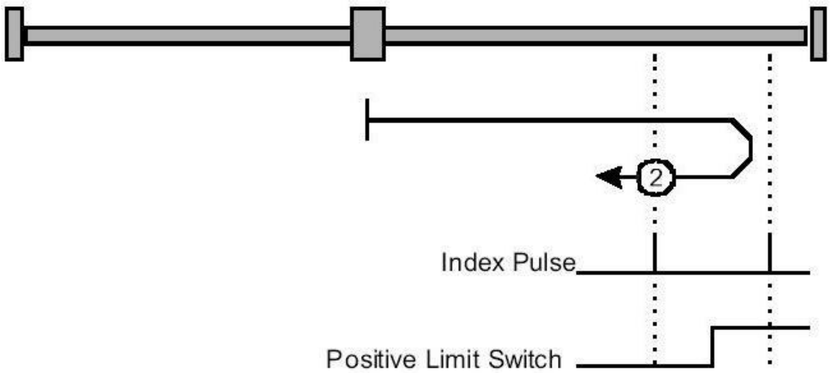

Figure 30: Homing on the positive limit switch and index pulse

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

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

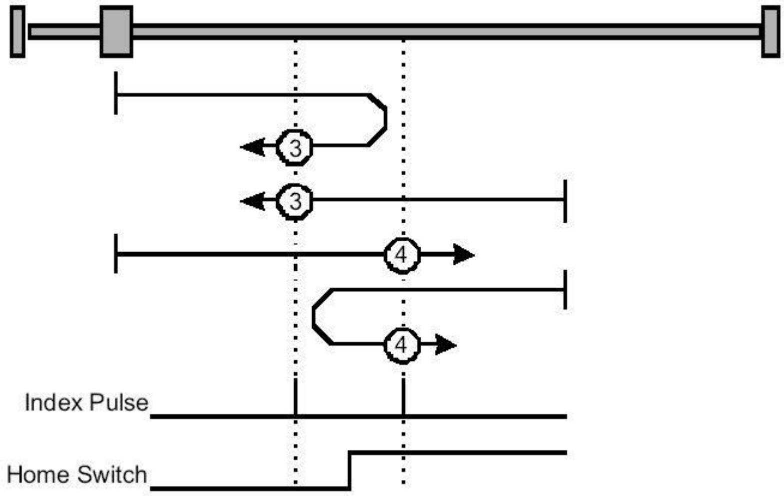

Figure 31: Homing on the positive home switch and index pulse

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

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

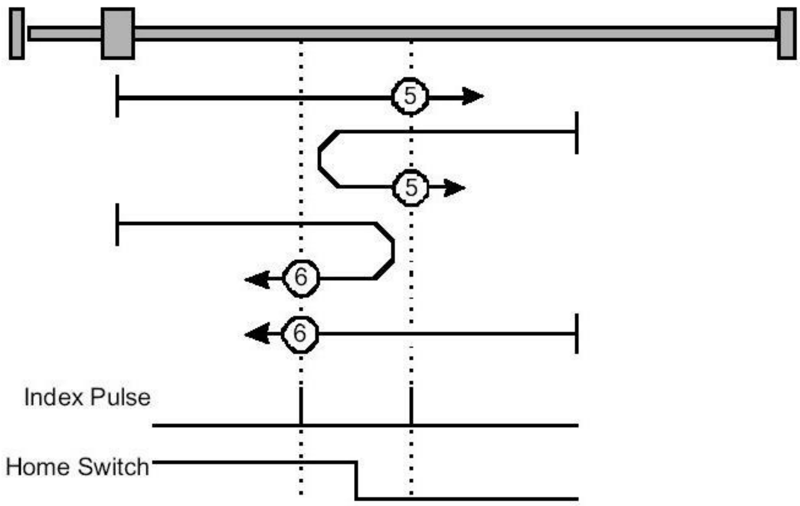

Figure 32: Homing on the negative home switch and index pulse

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

These methods use a home switch that is active over only a portion of the travel; in effect, the switch has a “momentary” action as the axle position sweeps past the switch.

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

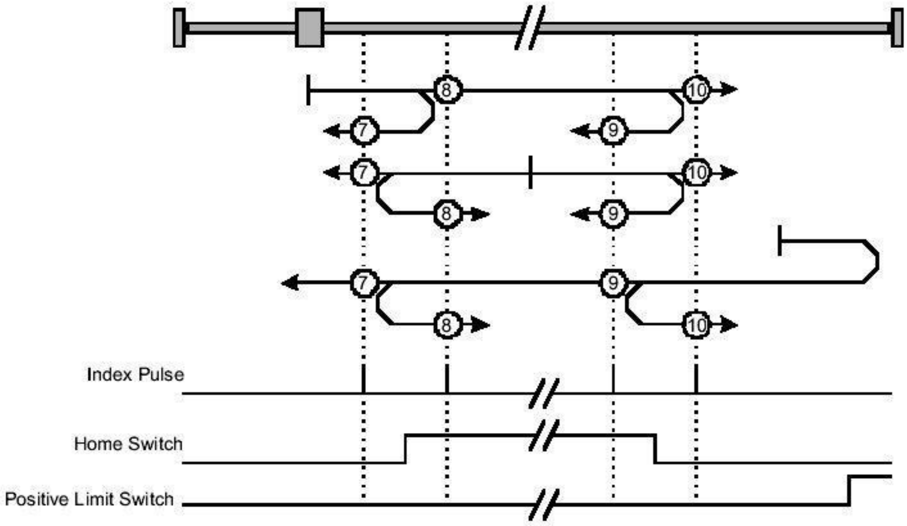

Figure 33: Homing on the home switch and index pulse — positive initial move

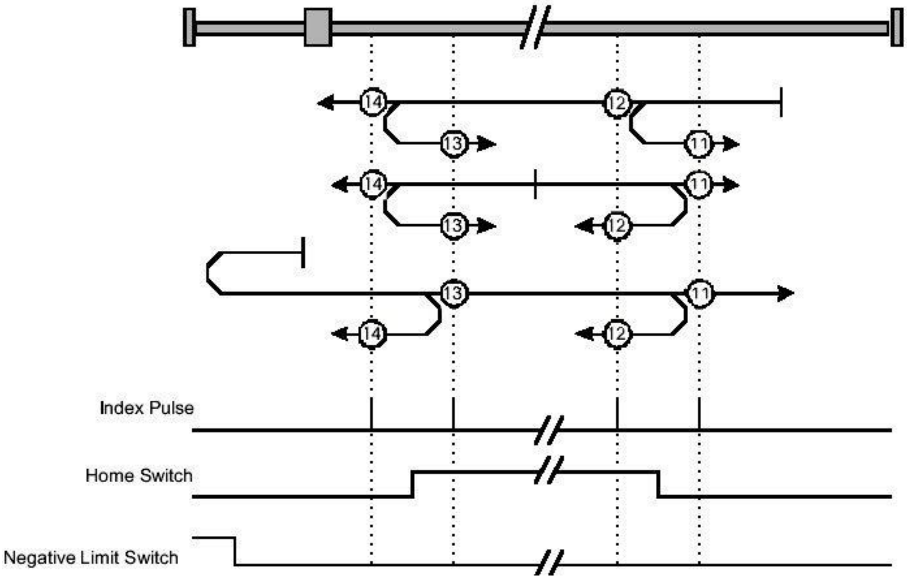

Figure 34: Homing on the home switch and index pulse — negative initial move

Methods 15 and 16: Reserved

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

Methods 17 to 30: Homing without an index pulse

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

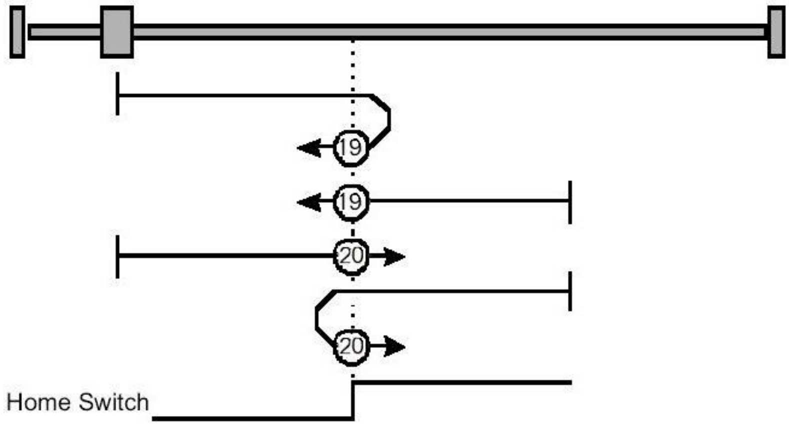

Figure 35: Homing on the positive home switch

Methods 31 and 32: Reserved

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

Methods 33 and 34: Homing on the index pulse

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

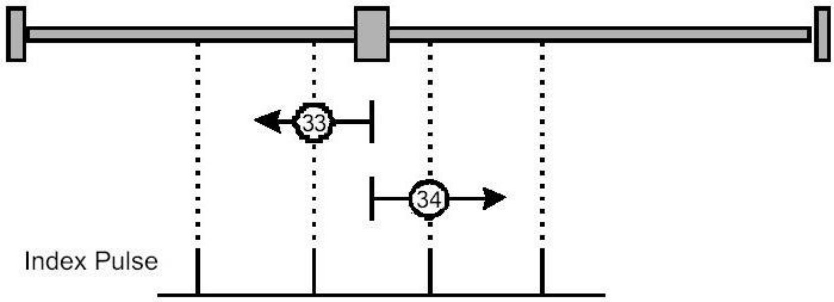

Figure 36: Homing on the positive home switch

Method 35 and 37: Homing on the current position

In these methods, the current position is taken to be the home position.

9.5.1 Method 35:

In this method, the current position shall be taken to be the home position.

9.5.2 Method 37:

In this method, the current position shall be taken to be the home position.

At the Home position (i.e. after homing process) the PositionActualValue_I32 (0x6064) is calculated as follows:

$$PositionActualValue\_I32\ (0x6064) = HomeOffset\_I32\ (0x607C)$$

10 Factor Group

10.1 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.

These units are used for all objects, which support user defined units (e.g. position actual value, profile velocity, profile acceleration). Objects, which values are not dependent on the factor group have fixed units specified with the objects.

The following default values shall be used for position, velocity and acceleration units. The default value for Pos Encoder Resolution should be used if PositionEncoderResolution_U32 0x608F is not available.

Parameter	Default Unit
Position Unit	inc
Velocity Unit	inc/s
Acceleration Unit	inc/s^2
Pos Encoder resolution	2^{32} inc/rev

Table 24 - Default values for units

10.2 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
$607E_h$	Polarity_U8	R	RW	VAR	UNSIGNED8	O
$608F_h$	PositionEncoderResolution_U32	R	RW	ARRAY	UNSIGNED32	O
6090_h	VelocityEncoderResolution_U32	R	RW	ARRAY	UNSIGNED32	O
6091_h	GearRatio_U32	R	RW	ARRAY	UNSIGNED32	O
6092_h	FeedConstant_U32	R	RW	ARRAY	UNSIGNED32	O
6093_h	PositionFactor_U32	R	RW	ARRAY	UNSIGNED32	O
6096_h	VelocityFactor_U32	R	RW	ARRAY	UNSIGNED32	O
6097_h	AccelerationFactor_U32	R	RW	ARRAY	UNSIGNED32	O
$60A2_h$	JerkFactor_U32	R	RW	ARRAY	UNSIGNED32	O

Table 25 - Object list factor groups

11 Endless positioning

The position data range (0x607A TargetPosition_I32, 0x6064 PositionActualValue_I32) and the machine relevant absolute position data of the axis sensor may differ. If the position data of an axis with infinite travel range (e.g. rotary axis, spindle etc.) were processed in absolute format, the axis would risk moving beyond the value range of the position data. This would lead to invalid position data; operating modes with position control would not be safe to operate.

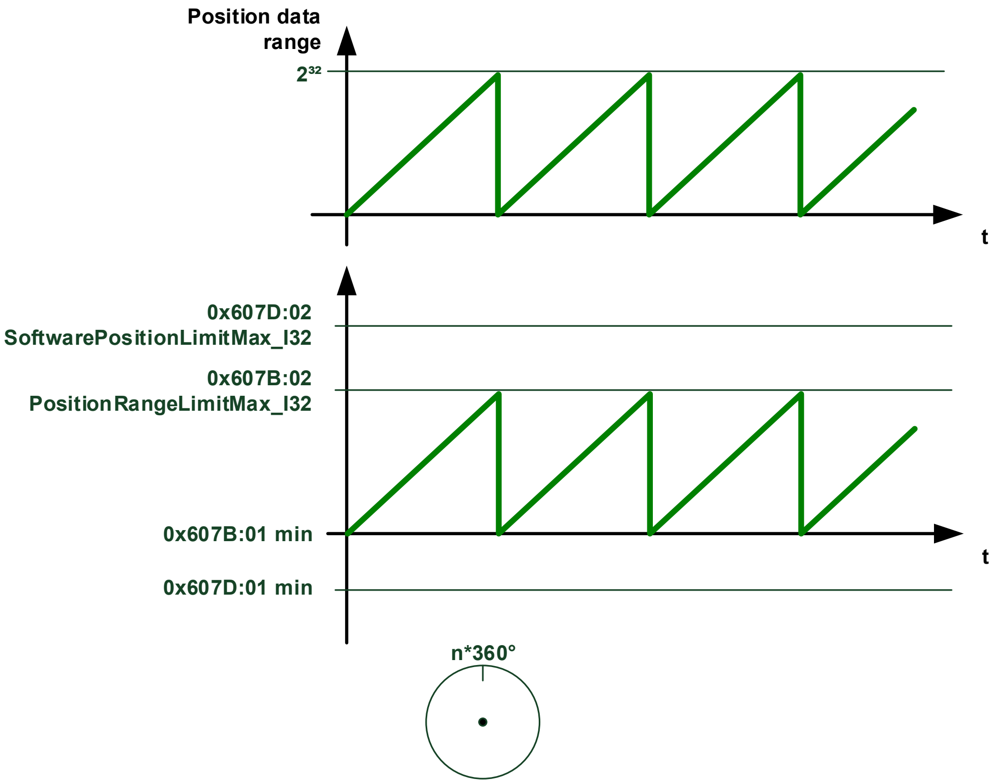

Figure 37: Position actual value in modulo format

The object 0x607B PositionRangeLimit_I32 indicates the maximum and minimum position range limits. It shall limit the range of the input value. On reaching or exceeding these limits, the input value shall wrap automatically to the other end of range. The SoftwarePositionLimit_I32 0x607D shall be out of the range of the position range limits.

To disable the position range limits for any reason the MinPositionRangeLimit_I32 (0x 607B:01) and MaxPositionRangeLimit_I32 (0x607B:02) shall be set to 0.

The drive shall always use the shortest way to the next TargetPosition_I32. Therefore the position command difference between two consecutive cycles may not exceed half the range:

Maximum position command difference = Position range / 2

Different movements are defined in object PositioningOptionCode_U16 60F2h in the bits 6 and 7.

Possible movements:
- normal positioning
- positioning only in negative direction
- positioning only in positive direction
- positioning with the shortest way

15	14 … 12	11 … 8	7	6	5	4	3	2	1	0
manufacturer specific	reserved	ip option	rado		request response option		change immediately option		relative option	

Figure 38: Object structure 0x60F2

The following table shows the bit definition of bit 6 and 7.

Bit 7	Bit 6	Definition
0	0	Normal positioning similar to linear axis If reaching or exceeding the position range limits (607B) the input value shall wrap automatically to the other end of the range
0	1	Positioning only in negative direction; if target position is higher than actual position, axis moves over "Min position limit" to target position
1	0	Positioning only in positive direction; if target position is lower than actual position, axis moves over "Max position limit" to target position
1	1	Positioning with the shortest way to the target position. Special condition: If the difference between actual value and target position in a 360° system is 180°, the axis will move in positive direction.

Table 26 - Value definition for bit 6 and 7 of object 60F2$_h$

The following figure depicts the movements depending on the settings of the rotary axis direction option bits. In this example min position range limit is 0 and max position range limit is 360.

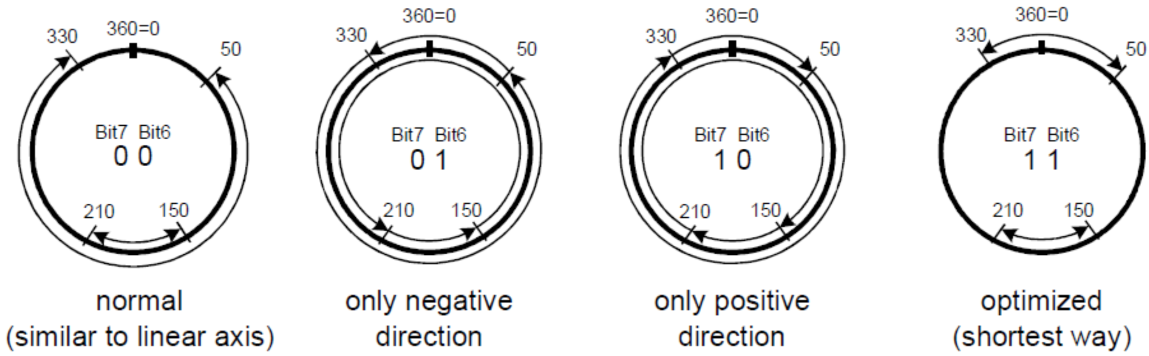

Figure 39: Rotary axis positioning

11.1 Object list

Index	Name	Category	Access	Object	Type	PDO - mapping
607B$_h$	PositionRangeLimit_I32	R	RW	ARRAY	INTEGER32	O
607D$_h$	SoftwarePositionLimit_I32	M	RW	ARRAY	INTEGER32	O
60F2$_h$	PositioningOptionCode_U16	M	RW	VAR	UNSIGNED16	O

Table 27 - Object list endless positioning

12 Object description

Object 1000h: DeviceType_U32

Index	1000_h	Object Type	VAR
Name	DeviceType_U32		
Data Type	UNSIGNED32	Category	M
Value Range	See Figure 2	Access	constant
Default Value	Manufacturer-specific	PDO Mapping	O

Object 6007h: AbortConnectionOptionCode_I16

Index	6007_h	Object Type	VAR
Name	AbortConnectionOptionCode_I16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	1	PDO Mapping	O

Object 603Fh: ErrorCode_U16

Index	$603F_h$	Object Type	VAR
Name	ErrorCode_U16		
Data Type	UNSIGNED16	Category	R
Value Range	UNSIGNED16	Access	RO
Default Value	No	PDO Mapping	O

Object 6040h: ControlWord_U16

Index	6040_h	Object Type	VAR
Name	ControlWord_U16		
Data Type	UNSIGNED16	Category	M
Value Range	UNSIGNED16	Access	RW
Default Value	Device and operation mode specific	PDO Mapping	M

Object 6041h: StatusWord_U16

Index	6041_h	Object Type	VAR
Name	StatusWord_U16		
Data Type	UNSIGNED16	Category	M
Value Range	UNSIGNED16	Access	RO
Default Value	No	PDO Mapping	M

Object 605Ah: QuickStopOptionCode_I16

Index	$605A_h$	Object Type	VAR
Name	QuickStopOptionCode_I16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	2	PDO Mapping	O

Object 605Bh: ShutdownOptionCode_16

Index	$605B_h$	Object Type	VAR
Name	ShutdownOptionCode_16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	0	PDO Mapping	O

Object 605Ch: DisableOperationOptionCode_I16

Index	$605C_h$	Object Type	VAR
Name	DisableOperationOptionCode_I16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	1	PDO Mapping	O

Object 605Dh: HaltOptionCode_I16

Index	$605D_h$	Object Type	VAR
Name	HaltOptionCode_I16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	1	PDO Mapping	O

Object 605Eh: FaultReactionOptionCode_I16

Index	$605E_h$	Object Type	VAR
Name	FaultReactionOptionCode_I16		
Data Type	INTEGER16	Category	O
Value Range	INTEGER16	Access	RW
Default Value	2	PDO Mapping	O

Object 6060h: ModesOfOperation_I8

Index	6060_h	Object Type	VAR
Name	ModesOfOperation_I8		
Data Type	INTEGER8	Category	M
Value Range	INTEGER8	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	M

Object 6061h: ModesOfOperationDisplay_I8

Index	6061_h	Object Type	VAR
Name	ModesOfOperationDisplay_I8		
Data Type	INTEGER8	Category	M
Value Range	INTEGER8	Access	RO
Default Value	No	PDO Mapping	M

Object 6064h: PositionActualValue_I32

Index	6064_h	Object Type	VAR
Name	PositionActualValue_I32		
Data Type	INTEGER32	Category	M
Value Range	INTEGER32	Access	RO
Default Value	No	PDO Mapping	M

Object 606Ch: VelocityActualValue_I32

Index	$606C_h$	Object Type	VAR
Name	VelocityActualValue_I32		
Data Type	INTEGER32	Category	M
Value Range	INTEGER32	Access	RO
Default Value	No	PDO Mapping	M

Object 6077h: TorqueActualValue_I16

Index	6077_h	Object Type	VAR
Name	TorqueActualValue_I16		
Data Type	INTEGER16	Category	R
Value Range	INTEGER16	Access	RO
Default Value	No	PDO Mapping	O

Object 607Ah: TargetPosition_I32

Index	$607A_h$	Object Type	VAR
Name	TargetPosition_I32		
Data Type	INTEGER32	Category	M (mode-specific)
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	M

Object 607Bh: PositionRangeLimit_I32

Index	$607B_h$	Object Type	ARRAY
Name	PositionRangeLimit_I32		
Data Type	INTEGER32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Min position range limit		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O
Sub-Index	02_h		
Name	Max position range limit		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 607Ch: HomeOffset_I32

Index	$607C_h$	Object Type	VAR
Name	HomeOffset_I32		
Data Type	INTEGER32	Category	M (mode-specific)
Value Range	INTEGER32	Access	RW
Default Value	0	PDO Mapping	O

Object 607Dh: SoftwarePositionLimit_I32

Index	$607D_h$	Object Type	ARRAY
Name	SoftwarePositionLimit_I32		
Data Type	INTEGER32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Min position limit		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O
Sub-Index	02_h		
Name	Max position limit		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 607Eh: Polarity_U8

Index	$607E_h$	Object Type	VAR
Name	Polarity_U8		
Data Type	UNSIGNED8	Category	R
Value Range	UNSIGNED8	Access	RW
Default Value	00_h	PDO Mapping	O

Object 6081h: ProfileVelocity_U32

Index	6081_h	Object Type	VAR
Name	ProfileVelocity_U32		
Data Type	UNSIGNED32	Category	M (mode-specific)
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	M

Object 6083h: ProfileAcceleration_U32

Index	6083_h	Object Type	VAR
Name	ProfileAcceleration_U32		
Data Type	UNSIGNED32	Category	M (mode-specific)
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 6084h: ProfileDeceleration_U32

Index	6084_h	Object Type	VAR
Name	ProfileDeceleration_U32		
Data Type	UNSIGNED32	Category	M (mode-specific)
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 6085h: QuickStopDeceleration_U32

Index	6085_h	Object Type	VAR
Name	QuickStopDeceleration_U32		
Data Type	UNSIGNED32	Category	R
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 608Fh: PositionEncoderResolution_U32

Index	$608F_h$	Object Type	ARRAY
Name	PositionEncoderResolution_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Encoder increments		
--	--	Category	M
Value Range	UNSIGNED32 (0 equals 2^{32})	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O
Sub-Index	02_h		
Name	Motor revolutions		
--	--	Category	M
Value Range	UNSIGNED32 (0 equals 2^{32})	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O

Object 6090h: VelocityEncoderResolution_U32

Index	6090_h	Object Type	ARRAY
Name	VelocityEncoderResolution_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Encoder increments per second		
--	--	Category	M
Value Range	UNSIGNED32 (0 equals 2^{32})	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O
Sub-Index	02_h		
Name	Motor revolutions per second		
--	--	Category	M
Value Range	UNSIGNED32 (0 equals 2^{32})	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O

Object 6091h: GearRatio_U32

Index	6091_h	Object Type	ARRAY
Name	GearRatio_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Motor shaft revolutions		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O
Sub-Index	02_h		
Name	Driving shaft revolutions		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O

Object 6092h: FeedConstant_U32

Index	6092_h	Object Type	ARRAY
Name	FeedConstant_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Feed		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O
Sub-Index	02_h		
Name	Shaft revolutions		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific (but not equal to 0)	PDO Mapping	O

Object 6096h: VelocityFactor_U32

Index	6096_h	Object Type	ARRAY
Name	VelocityFactor_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Numerator		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O
Sub-Index	02_h		
Name	Divisor		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O

Object 6097h: AccelerationFactor_U32

Index	6097_h	Object Type	ARRAY
Name	AccelerationFactor_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Numerator		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O
Sub-Index	02_h		
Name	Divisor		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O

Object 6098h: HomingMethod_I8

Index	6098_h	Object Type	VAR
Name	HomingMethod_I8		
Data Type	INTEGER8	Category	M (mode-specific)
Value Range	INTEGER8	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 6099h: HomingSpeed_U32

Index	6099_h	Object Type	ARRAY
Name	HomingSpeed_U32		
Data Type	UNSIGNED32	Category	M (mode-specific)
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Speed during search for switch		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O
Sub-Index	02_h		
Name	Speed during search for zero		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 609Ah: HomingAcceleration_U32

Index	$609A_h$	Object Type	VAR
Name	HomingAcceleration_U32		
Data Type	UNSIGNED32	Category	M (mode-specific)
Value Range	UNSIGNED32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 60A2h: JerkFactor_U32

Index	$60A2_h$	Object Type	ARRAY
Name	JerkFactor_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Numerator		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O
Sub-Index	02_h		
Name	Divisor		
--	--	Category	M
Value Range	1 to 2^{32}-1	Access	RW
Default Value	1	PDO Mapping	O

Object 60B0h: PositionOffset_I32

Index	$60B0_h$	Object Type	VAR
Name	PositionOffset_I32		
Data Type	INTEGER32	Category	R
Value Range	INTEGER32	Access	RW
Default Value	0	PDO Mapping	O

Object 60B1h: VelocityOffset_I32

Index	$60B1_h$	Object Type	VAR
Name	VelocityOffset_I32		
Data Type	INTEGER32	Category	R
Value Range	INTEGER32	Access	RW
Default Value	0	PDO Mapping	O

Object 60B2h: TorqueOffset_I16

Index	$60B2_h$	Object Type	VAR
Name	TorqueOffset_I16		
Data Type	INTEGER16	Category	R
Value Range	INTEGER16	Access	RW
Default Value	0	PDO Mapping	O

Object 60C0h: InterpolationSubModeSelect_I16

Index	$60C0_h$	Object Type	VAR
Name	InterpolationSubModeSelect_I16		
Data Type	INTEGER16	Category	R
Value Range	INTEGER16	Access	RW
Default Value	0000_h	PDO Mapping	O

Object 60C1h: InterpolationDataRecord_I32

Index	$60C1_h$	Object Type	ARRAY
Name	InterpolationDataRecord_ I32		
Data Type	INTEGER32	Category	M (mode-specific)
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	01_h to FE_h	Access	constant
Default Value	Manufacturer-specific	PDO Mapping	No
Sub-Index	01_h		
Name	1st set-point		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	RPDO
Sub-Index	02_h to FE_h		
Name	2nd - 254th set-point		
--	--	Category	M
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	O

Object 60C2h: InterpolationTimePeriod_U8

Index	$60C2_h$	Object Type	ARRAY
Name	InterpolationTimePeriod_U8		
Data Type	UNSIGNED8	Category	M (mode specific)
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Intepolation time period value		
--	--	Category	M
Value Range	UNSIGNED8	Access	RW
Default Value	01_h	PDO Mapping	O
Sub-Index	02_h		
Name	Interpolation time index		
--	--	Category	M
Value Range	-128 to +63	Access	RW
Default Value	-3	PDO Mapping	O

Object 60C4h: InterpolationDataConfiguration_U32

Index	$60C4_h$	Object Type	ARRAY
Name	InterpolationDataConfiguration_U32		
Data Type	UNSIGNED32	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	02_h	Access	constant
Default Value	02_h	PDO Mapping	No
Sub-Index	01_h		
Name	Maximum buffer size		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	No	PDO Mapping	O
Sub-Index	02_h		
Name	Actual buffer size		
--	--	Category	M
Value Range	UNSIGNED32	Access	RW
Default Value	0000 0000h	PDO Mapping	O

Object 60E3h: SupportedHomingMethods_I8

Index	$60E3_h$	Object Type	ARRAY
Name	SupportedHomingMethods_I8		
Data Type	INTEGER8	Category	R
Sub-Index	00_h		
Name	Highest sub-index supported		
--	--	Category	M
Value Range	01_h to FE_h	Access	constant
Default Value	No	PDO Mapping	No
Sub-Index	01_h to FE_h		
Name	1st supported homing method		
--	--	Category	M
Value Range	INTEGER8	Access	RO
Default Value	No	PDO Mapping	O
Sub-Index	02_h to FE_h		
Name	2nd to 254th supported homing method		
--	--	Category	O
Value Range	INTEGER8	Access	RO
Default Value	Manufacturer-specific	PDO Mapping	O

Object 60F2h: PositioningOptionCode_U16

Index	$60F2_h$	Object Type	VAR
Name	PositioningOptionCode_U16		
Data Type	UNSIGNED16	Category	R
Value Range	UNSIGNED16	Access	RW
Default Value	0000_h	PDO Mapping	O

Object 60FFh: TargetVelocity_I32

Index	$60FF_h$	Object Type	VAR
Name	TargetVelocity_I32		
Data Type	INTEGER32	Category	M (mode-specific)
Value Range	INTEGER32	Access	RW
Default Value	Manufacturer-specific	PDO Mapping	M

Object 6502h: SupportedDriveModes_U32

Index	6502_h	Object Type	VAR
Name	SupportedDriveModes_U32		
Data Type	UNSIGNED32	Category	R
Value Range	UNSIGNED32	Access	RO
Default Value	No	PDO Mapping	O