Servodyn-D / Servodyn-M

Servodyn-D, Servodyn-M with CANopen Parameter manual

Edition 101

BOSCH
Automation

Servodyn-D / Servodyn-M

Servodyn-D, Servodyn-M with CANopen Parameter manual

1070 066 095-101 (01.07) GB

Contents

1 Safety Instructions

Please use this manual to commission Servodyn-D or Servodyn-M with CANopen. Store this manual in a place to which all users have access at any time.

1.1 Intended use

This manual contains information required for the intended use of this product.

The drive inverters described

- have been developed, manufactured, tested and documented in compliance with the safety standards. These products pose no danger to persons or property if they are used in accordance with the handling stipulations and safety notes prescribed for their configuration, mounting, and proper operation.
- comply with the requirements of
  - the EMC Directives (89/336/EEC, 93/68/EEC and 93/44/EEC)
  - the EMC product standard EN 61800-3
  - the Low-Voltage Directive (73/23/EEC)
  - the harmonized standards EN 50178 (VDE 0160) and EN 60146-1-1 (VDE 0558-11)
- are designed for operation in industrial environments (emission class A), i.e.
  - no direct connection to public low-voltage power supply,
  - connection to the medium- or high-voltage system via a transformer.

  In residential environments, in trade and commerce as well as small enterprises class A equipment may only be used if it does not inadmissibly interfere with other equipment.

☞ **This is a class A device which may cause radio interference in residential environments. In this case, the operator may be required to take suitable countermeasures and to bear the cost of the same.**

Before putting the drive inverters into operation, ensure that the machine which the inverters are to be installed in meets the stipulations of the machinery directive (98/37/EEC, 98/79/EEC) and the EMC directive (89/336/EEC).

The faultless, safe functioning of the product requires proper transport, storage, erection and installation as well as careful operation.

1.2 Qualified personnel

The requirements as to qualified personnel depend on the qualification profiles described by ZVEI (Zentralverband Elektrotechnik und Elektronikindustrie – German Electrical and Electronic Manufacturers' Association) and the VDMA (Verband deutscher Maschinen- und Anlagenbau – German Engineering Federation) in:
Weiterbildung in der Automatisierungstechnik
edited by: ZVEI and VDMA
MaschinenbauVerlag
Postfach 71 08 64
60498 Frankfurt/Germany

The present manual is designed for **drive technicians**. They need special knowledge on drive systems and the CANopen bus.

Programming, start and operation as well as the modification of program parameters is reserved to properly trained personnel! This personnel must be able to judge potential hazards arising from programming, program changes and in general from the mechanical, electrical, or electronic equipment.

Interventions in the hardware and software of our products, unless described otherwise in this manual, are reserved to our specialized personnel.

Tampering with the hardware or software, ignoring warning signs attached to the components, or non-compliance with the warning notes given in this manual may result in serious bodily injury or material damage.

Only electrotechnicians as recognized under IEV 826-09-01 (modified) who are familiar with the contents of this manual may install and service the products described.

Such personnel are

- those who, being well trained and experienced in their field and familiar with the relevant norms, are able to analyze the jobs being carried out and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through years of experience that would normally be acquired through formal technical training.

With regard to the foregoing, please note our comprehensive range of training courses. Our training center will be pleased to provide you with further information,
telephone: (+49) (0 60 62) 78-258.

1.3 Safety markings on products

Warning of dangerous electrical voltage!

Electrostatically sensitive components!

Lug for connecting PE conductor only!

Connection of shield conductor only

1.4 Safety instructions in this manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of a **dangerous electrical voltage.** The failure to observe the instructions in this manual in whole or in part may result in **personal injury**.

DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.

CAUTION
This symbol is used wherever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

☞ This symbol is used to draw the user's attention to special circumstances.

★ This symbol is used if user activities are required.

1.5 Safety instructions concerning the product described

DANGER
Danger of life through inadequate EMERGENCY-STOP devices!
EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system!
First check the EMERGENCY-STOP circuit, then switch the system on!

DANGER
Danger for persons and equipment!
Test every new program before starting up a system!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!
The consequences may include severe injury, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Bosch.

DANGER
Health hazards through destroyed electrical components!
Do not destroy any built-in components. Dispose of destroyed components in a proper manner.

DANGER
Please note your local, system-specific regulations and requirements as well as the proper use of tools, hoisting and transport equipment as well as the applicable standards, regulations, and accident prevention regulations.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance works must be performed on inactive systems! The system must be protected against unauthorized or accidental reclosing.

Measuring or test activities on the live system are reserved to qualified electrical personnel!

DANGEROUS ELECTRICAL VOLTAGE
Lethal voltages of up to 375 V DC against ground on all power connections and DC link connections!

The drives must not be switched on unless all covers have been fitted! When the drive has been disconnected from mains, wait for up to 5 minutes until the system is de-energized before removing any covers.

The drive must always be examined for safe isolation from supply!

CAUTION
Use only spare parts approved by Bosch!

CAUTION
Danger to the module!
All ESD protection measures must be observed when using the module! Prevent electrostatic discharges!

The following protective measures must be observed for modules and components sensitive to electrostatic discharge (ESD)!

- Personnel responsible for storage, transport, and handling must have training in ESD protection.
- ESD-sensitive components must be stored and transported in the prescribed protective packaging.
- ESD-sensitive components may only be handled at special ESD-workplaces.
- Personnel, working surfaces, as well as all equipment and tools which may come into contact with ESD-sensitive components must have the same potential (e.g. by grounding).
- Wear an approved grounding bracelet. The grounding bracelet must be connected with the working surface through a cable with an integrated 1 MΩ resistor.
- ESD-sensitive components may by no means come into contact with chargeable objects, including most plastic materials.
- When ESD-sensitive components are installed in or removed from equipment, the equipment must be de-energized.

1.6 Documentation, software release and trademarks

Documentation

The present manual provides information about CANopen objects for Servodyn-D and Servodyn-M.

Overview of available manuals:

Manuals	Part no.			
	German	English	French	Italian
Servo motors SF, SR	1070 066 004	1070 066 024	1070 066 048	1070 066 046
Asynchronous motors DU	1070 066 007	1070 066 027	–	–
Configuration - Manual for overview and rating	1070 066 009	1070 066 029	1070 066 059	1070 066 049
Servodyn-D, Interface conditions	1070 066 010	1070 066 030	1070 066 060	1070 066 050
Servodyn-D, Interface conditions - Stand alone version	1070 066 016	1070 066 036	1070 066 066	1070 066 056
Servodyn-D, - Parameter manual (without CANopen)	1070 066 018	1070 066 038	1070 066 068	1070 066 058
Servodyn-D, Servodyn-M - Parameter manual CANopen	1070 066 094	1070 066 095	–	–
Servodyn-D with SERCOS interface - Parameter and commissioning manual	1070 066 011	1070 066 031	–	1070 066 051
Servodyn-D with analog interface - Commissioning manual	1070 066 014	1070 066 034	–	–
Servodyn-D with CANrho interface - Commissioning manual	1070 066 017	1070 066 037	–	–
Servodyn-D with motion control - Commissioning manual	1070 066 015	1070 066 035	–	–
Servodyn-D with PROFIBUS-DP - Commissioning manual	1070 066 090	1070 066 091	–	–
Diagnostics, maintenance	1070 066 012	1070 066 032	1070 066 062	1070 066 052
RSU, Redundant safety monitoring	1070 066 006	1070 066 026	1070 066 081	1070 066 082
EMC manual	1070 066 072	1070 066 074	1070 066 075	1070 066 076
External load switching module	1070 066 077	1070 066 080	–	–

Release

☞ **The present manual applies to the following releases:**
DM/DS software: 0.001 or higher

- The current software release number can be viewed by selecting parameter S-0-0030 with the DSS-D Commissioning and Service System, or in the ”Software” field of the module configuration display (DIAGNOSTICS ▶ MODULE CONFIGURATION).
- For information concerning the current DSS software release, refer to HELP ▶ ABOUT...
- The current VM software release can only be read from the 7-segment display during test operation. For this purpose, turn dip switch ”T” on the VM’s personality module ”on”:

The following appears in a running, flashing display:
“**Cxx.ZZ.ddmmyyyy**“

Where:
xx = software release number
ZZ = (internal)
dd = software creation day
mm = software creation month
yyyy = software creation year

Trademarks

All trademarks of software installed on Bosch products upon delivery are the property of the respective manufacturer.

Upon delivery, all installed software is copyright-protected. The software may only be reproduced with the approval of Bosch or in accordance with the license agreement of the respective manufacturer.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corporation.

2 Basic Architecture

2.1 Data Objects

The CANopen data transfer is performed using data objects.
The drive contains:

- Process Data Objects (PDO) and
- Service Data Objects (SDO)

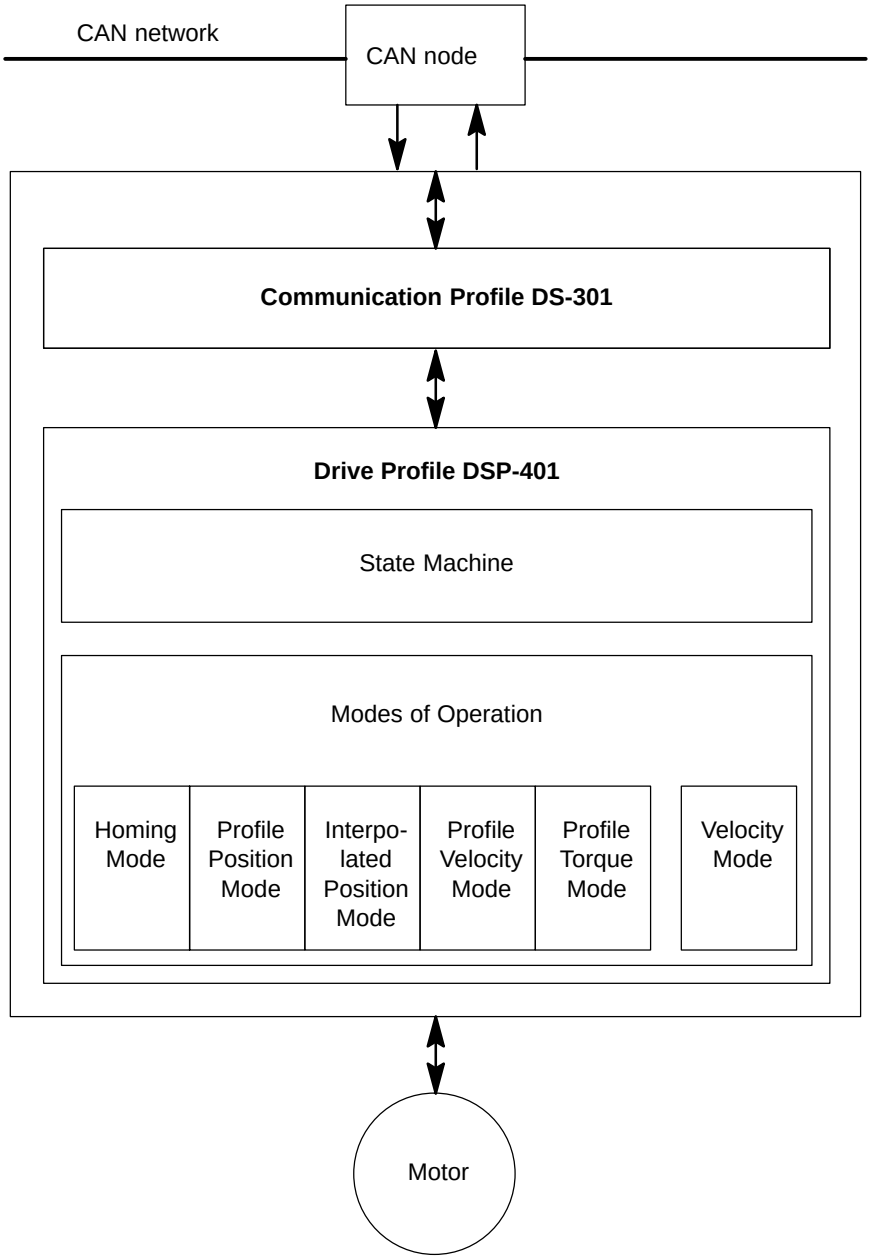
Process Data Objects (PDO):

PDOs are used for the fast transfer of real-time data to and from the drive. The transfer is performed without handshake and therefore requires no protocol overhead. Eight application data bytes can be transported in one CAN frame.
PDOs are determined by entries in the object dictionary, as described in chapter 5 to 14. The data types of the object dictionary are explained in chapter 3.

Service Data Objects (SDO):

SDOs are used for data transfer with a kind of handshake. They are used for the access to entries in the object dictionary.
Especially the configuration of the drive behaviour is carried out by these objects.

2.2 Communication Architecture



Drive Function Control

The starting and stopping of the drive and several mode specific commands are executed by the State Machine, see section 6.2. All mode specific actions are described in chapter 8 to 14.

Modes of Operation

The operation mode defines the behaviour of the drive. The following operation modes are defined in the drive profile:

- **Homing Mode** (see chapter 9)
  In this mode of operation, it is possible to choose between various methods to find a home position.

- **Profile Position Mode** (see chapter 8)
  The positioning of the drive is defined in this mode. Speed, position and acceleration can be limited. Corresponding movements are possible using a trajectory generator.

- **Interpolated Position Mode** (see chapter 11)
  This mode provides for the time interpolation of single axles and the spatial interpolation of several co-ordinated axles.

- **Profile Velocity Mode** (see chapter 12)
  In this mode, the velocity of the drive is specified with no special regard to the position. It supplies limit functions and contains a trajectory generator.

- **Profile Torque Mode** (see chapter 13)
  In this mode of operation, the drive is operated torque controlled.

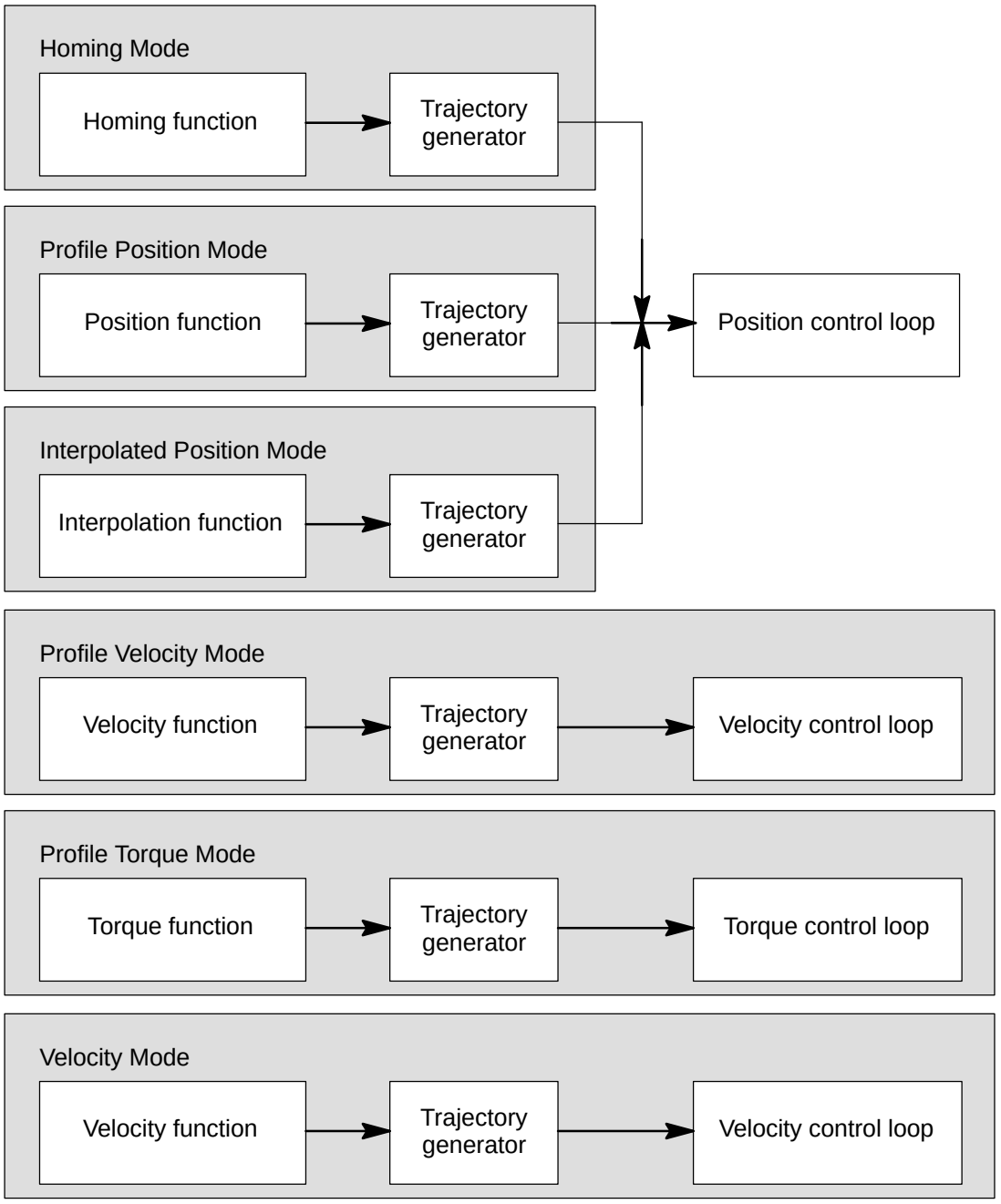
- **Velocity Mode** (see chapter 14)
  Many frequency inverters use this simple mode to control the velocity of the drive with limits and ramp functions.

☞ **The Velocity Mode is separated from the other modes and does not interfere with them so much. For this reason, the naming of object dictionary entries differs a little bit from the other modes.**

Trajectory Generator

The chosen operation mode and the corresponding parameters (objects) define the input values of the Trajectory Generator. The trajectory generator supplies its values to the control loop (see figure).

Functional Architecture



3 Operating Principle

3.1 Introduction

With the Drive Profile, the drives are assigned an understandable and unique behaviour on the CAN network. The CANopen Device Profile is built on top of a CAN communication profile, called CANopen, describing the basic communication mechanisms common to all devices at the CAN network.

Configuration information, normally via service data objects for I/O configurations, limit parameters or application specific parameters are transferred to the drive through the CAN bus. During operation, data can be obtained from the drive via CAN bus by either polling or event driven in case of an interruption.

3.2 Standardisation via Profiling

Backed by the drive profile, the communication capability of the drives through CANopen are standardised and thus improved.
Operating devices from different manufacturers together on one CAN bus is possible at low expenditure.

The profile defines a ”standard drive” with basic functions supported by every drive. This basic functionality ensures that at least simple non-manufacturer-specific operation of a device is possible.
The standard drive for example provides the ’Quick Stop’ function. Consequently, it is possible to stop any drive unit which is supported by the CANopen drive profile using the same command.

The standard drive is extended by the optional functionality defined within the standardised drive profile. Such optional functionality does not have to be implemented in all drives. However, if a function of this kind has been implemented, this must be done in the same fixed manner.

In addition to this, Servodyn drives contain very specific functionality which support their performance capacity.
Due to the mandatory drive characteristics, basic network operation is guaranteed at all times.

3.3 Object Dictionary

3.3.1 Object Areas

The most important part of a drive profile is the object dictionary description. The object dictionary is a grouping of objects accessible via the network. Each object is addressed using a 16-bit index so that the object dictionary may contain a maximum of 65536 entries.

The layout of the standard object dictionary is shown below:

Index (hex)	Object
0000	not used
0001 – 001F	Static Data Types
0020 – 003F	Complex Data Types
0040 – 005F	Manufacturer Specific Data Types
0060 – 0FFF	Reserved
1000 – 1FFF	Communication Profile Area
2000 – 5FFF	Manufacturer Specific Profile Area
6000 – 9FFF	Standardised Device Profile Area
A000 – FFFF	Reserved

Static Data Types (index 0001_h through $001F_h$):
They contain type definitions for standard data types like boolean, integer, floating point, string, etc. These entries are included for reference only, they cannot be read or written.

Complex Data Types (index 0020_h through $003F_h$)
Pre-defined structures that are composed of standard data types and are common to all devices.

Manufacturer Specific Data Types (index 0040_h through $005F_h$)
Structures that are composed of standard data types but are specific to a particular device.

Communication Profile Area (index 1000_h through $1FFF_h$)
It contains the parameters for the communication profile on the CAN network. These entries are common to all devices.

Manufacturer Specific Profile Area (index 2000_h through $5FFF_h$)
It contains manufacturer specific parameters which are provided additionally.

3.3.2 Usage of Index and Sub-index

The complete 16-bit index addresses all entries within the object dictionary. In case of a simple variable this references the value of this variable directly. In case of records and arrays however, the index addresses the whole data structure.

The sub-index is used to access individual elements of structures of data via the network:

- For simple dictionary entries such as Unsigned8, Boolean, Integer32 the value for the sub-index is always zero.
- For complex object dictionary entries such as ARRAY or RECORD, the sub-index refers to fields within a data structure pointed to by the main index.

Sub-index counting starts with one. For example in the chapter Factor Group exists the object $608F_h$ named position_encoder_resolution. Because this may be a fraction, two integers in an array are used to describe this object. The drive uses the two values in the following manner:

$$\text{position\_encoder\_resolution} = \frac{\text{encoder\_increments}}{\text{motor\_resolutions}}$$

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

Main Index	Sub-index	Variable Accessed	Data type
648F	0	Number of elements	Unsigned8
	1	encoder_increments	Unsigned32
	2	motor_revolutions	Unsigned32

3.3.3 Entry Representation

In the following chapters, the abbreviations below will be used to represent the object dictionary entries in the tables:

Meaning of the Table Sequences:		
Index	16-bit index from the object dictionary to represent a function, task or data	
Name	defined name of the object	
Object code	type of representation in the object, e.g. VAR, ARRAY, RECORD etc. (RECORD = manufacturer specific)	
Data type	Type of information presentation, e.g. Unsigned32 (without sign, 32 bit) etc.	
Object class	M: object mandatory for drives O: object optional and dependent on mode of operation	
Access	Type of access: ro: read only wo: write only rw: read and write	
PDO Mapping	determines whether the object can be transmitted cyclically with PDOs: No: no mapping, i.e. no cyclical transmission Yes: mapping permissible, i.e. is transmitted cyclically	
Units	physical units for the object value	
Value range	permissible value range of the object	
Default value	preassigned standard value of the object	
Abbreviations for modes of operation:		
pp	Profile Position Mode	m : mandatory o : optional – : not used
pv	Profile Velocity Mode	
vl	Velocity Mode	
hm	Homing Mode	
ip	Interpolated Position Mode	
tq	Profile Torque Mode	
all	Mandatory for all modes	
Chapter abbreviations of names		
ce	Common object dictionary entries	
dc	Device control	Device control
pc	mandatory (m), optional (o) or not used (–) for position control function	

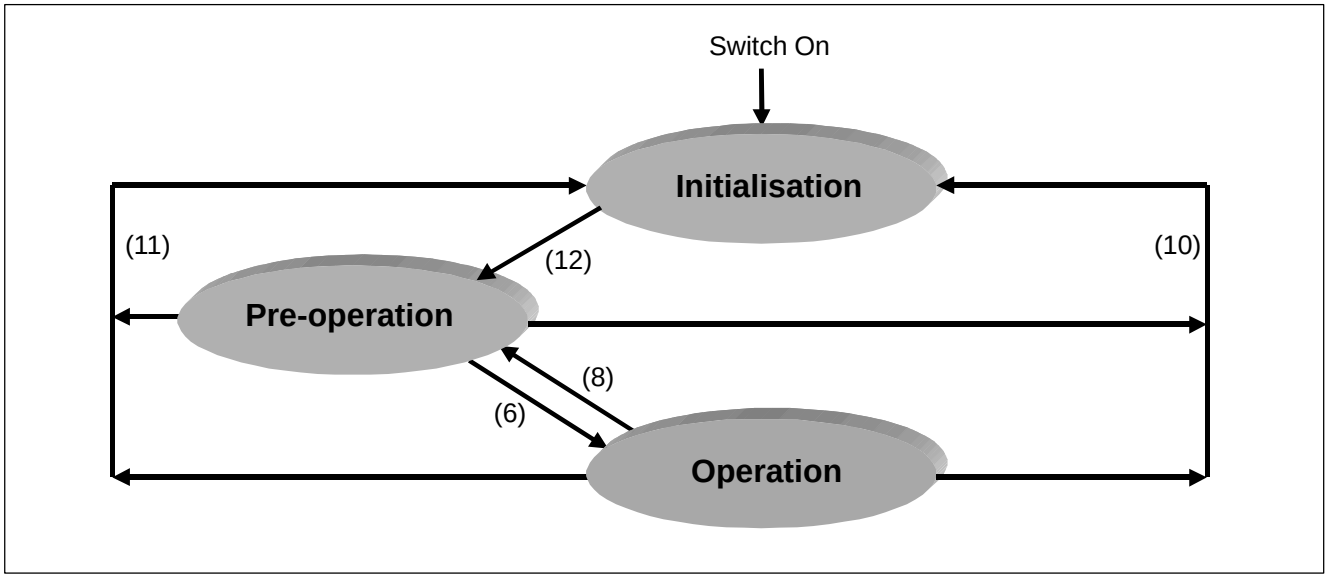
4 Telegram Structure

4.1 Standard Structure

The CANopen telegrams of Servodyn drives are in accordance with CAN in the Automation Draft Standard 301 (CiA DS-301, version 4.01 of June 2000).

4.1.1 State Diagram

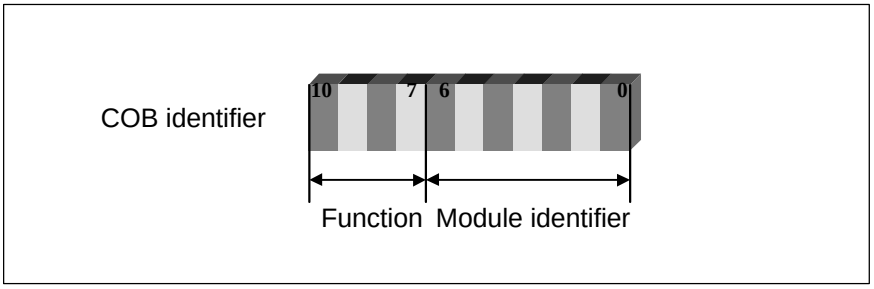
States according to CiA DS-301, page 8-10 (minimum performance device):



(6) “start_remote_node”
(8) “enter_pre-operational_state”
(10) “reset_node”
(11) “reset_communication”
(12) Initialisation completed – automatically in pre-operational state

4.1.2 Communication Object (COB) Identifier

For a detailed description, see ”Network Management and Identifier Distribution” in CiA DS-301, page 9-55.



Requested objects

Object	Function code (Bin)	COB identifier (Hex)	Servodyn-D, -M (Hex)	CMS priority group*	Communication parameter at index
NMT (network management)	0000	0x000	0x000	0	–
SYNC	0001	0x080	0x080	0	1005

* **CMS**: CAN-based message specification

Peer-to-peer objects

Object	Function code (Bin)	COB identifier (Hex)	Servodyn-D, -M (Hex)	CMS priority group*	Communication parameter at index
EMCY (Emergency object)	0001	0x081 – 0x0FF	0x080 + Address switch	0, 1	–
PDO1 (tx) (Process data object)	0011	0x181 – 0x1FF	0x180 + Address switch	1, 2	1800 1A00
PDO1 (rx) (Process data object)	0100	0x201 – 0x27F	0x200 + Address switch	2	1400 1600
PDO2 (tx) (Process data object)	0101	0x281 – 0x2FF	0x280 + Address switch	2, 3	1801 1A01
PDO2 (rx) (Process data object)	0110	0x301 – 0x37F	0x300 + Address switch	3, 4	1401 1601
SDO (tx) (Service data object)	1011	0x581 – 0x5FF	0x580 + Address switch	6	
SDO (rx) (Service data object)	1100	0x601 – 0x67F	0x600 + Address switch	6, 7	

4.1.3 Baud Rate

The baud rate is set using rotary switch S4 on the front panel of the Servodyn-D drives:

Rotary switch S4										
Switch position	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9...F**
Baud rate [Kbit/s]	1000	800	500	250	125	100	50	20	10	1000

4.2 Network Management (NMT)

For a detailed description, see ”Network Management and Identifier Distribution” in CiA DS-301, page 9-41.

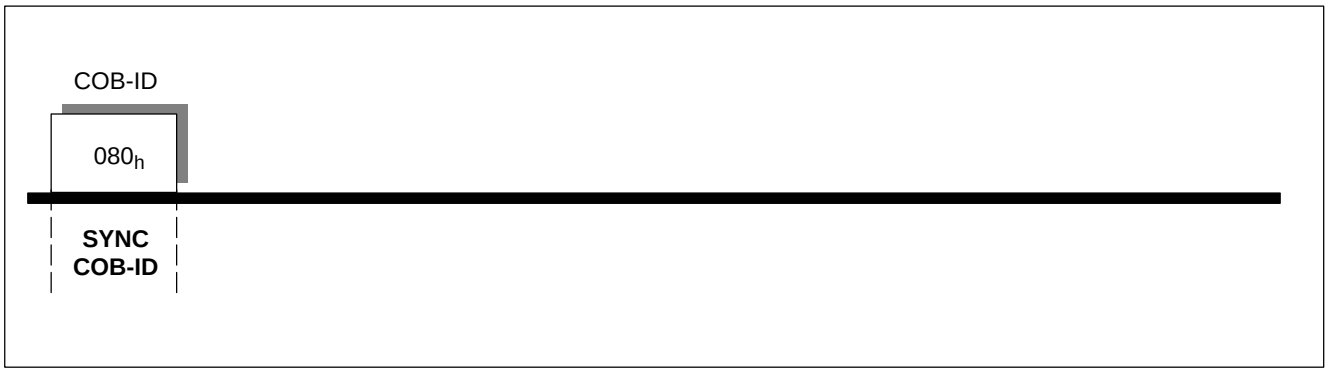
NMT Message

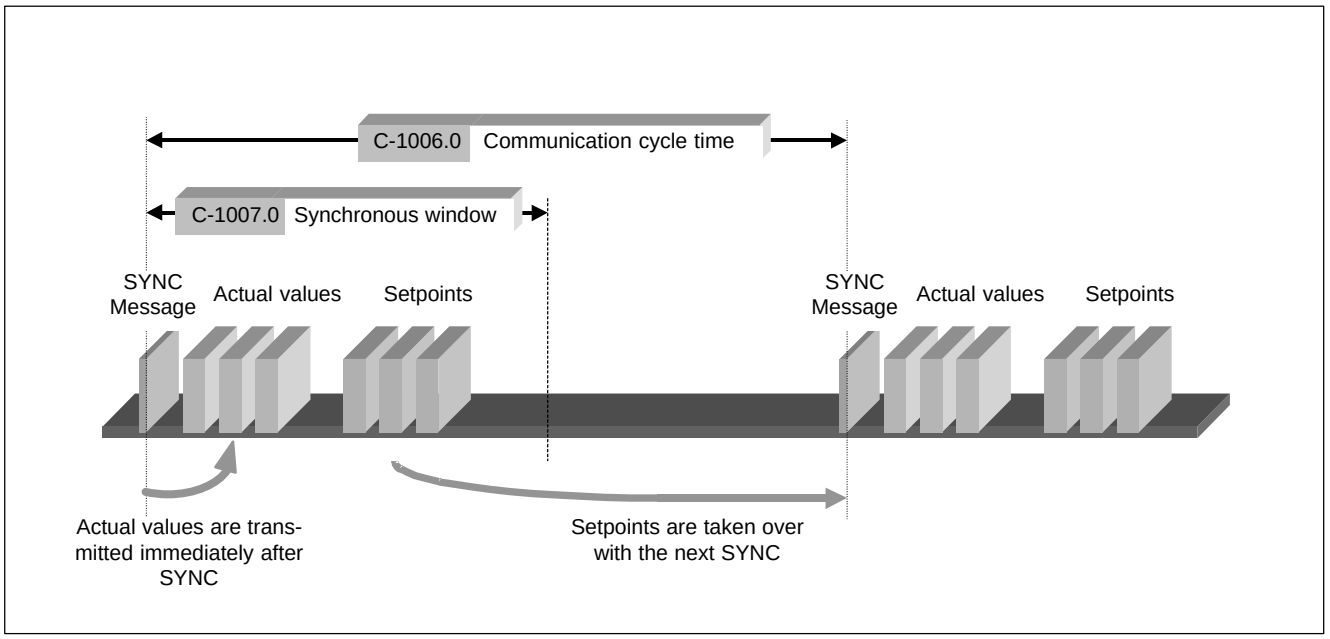
n	User ID of the NMT slave
0	All NMT slaves

1	“start_remote_node” service	module display: 4
2	“stop_remote_node” service	module display: 2
3	“disconnect_remote_node” service	module display: 2
128	“disconnect_remote_node” service	module display: 2
129	“reset_node” service	module display: 0
130	“reset_communication” service	module display: 0

4.3 Synchronisation

4.3.1 SYNC Message





4.3.2 Cyclical Setpoint/Actual Value Processing

C-2007$_h$ feedback_acquisition_starting_time

Time of measurement of the actual values at the end of the SYNC telegram. The same value should be set for all axles, so that synchronous measurement of all values is ensured.

Index	**C-2007$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	1000 µs	
Value range	–	
Default value	1	

C-2008$_h$ command_valid_time

Setpoint valid time after the end of the SYNC telegram. The value may be the same for all axles.

Index	**C-2008$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	1000 µs	
Value range	–	
Default value	1	

The time C-2008 ”command_valid_time” validates the **setpoint** and thus determines after what time after the end of the SYNC telegram the drive may access the new setpoints. The master may thus specify the same time for setpoint valid in the 1 ms grid for all drives co-ordinated with each other.

Using C-2007 ”feedback_acquisition_starting_time”, the master may specify the same time for **actual values** in the 1 ms grid for all drives co-ordinated with each other. This ensures synchronisation of the actual value capture in the drives in question.
Using these parameters, it is possible to reduce the dead time for the control.

As the drive transmits the PDOtx immediately after receiving the SYNC telegram, it is favourable to set the measuring time of the actual value as close as possible before the next SYNC telegram.

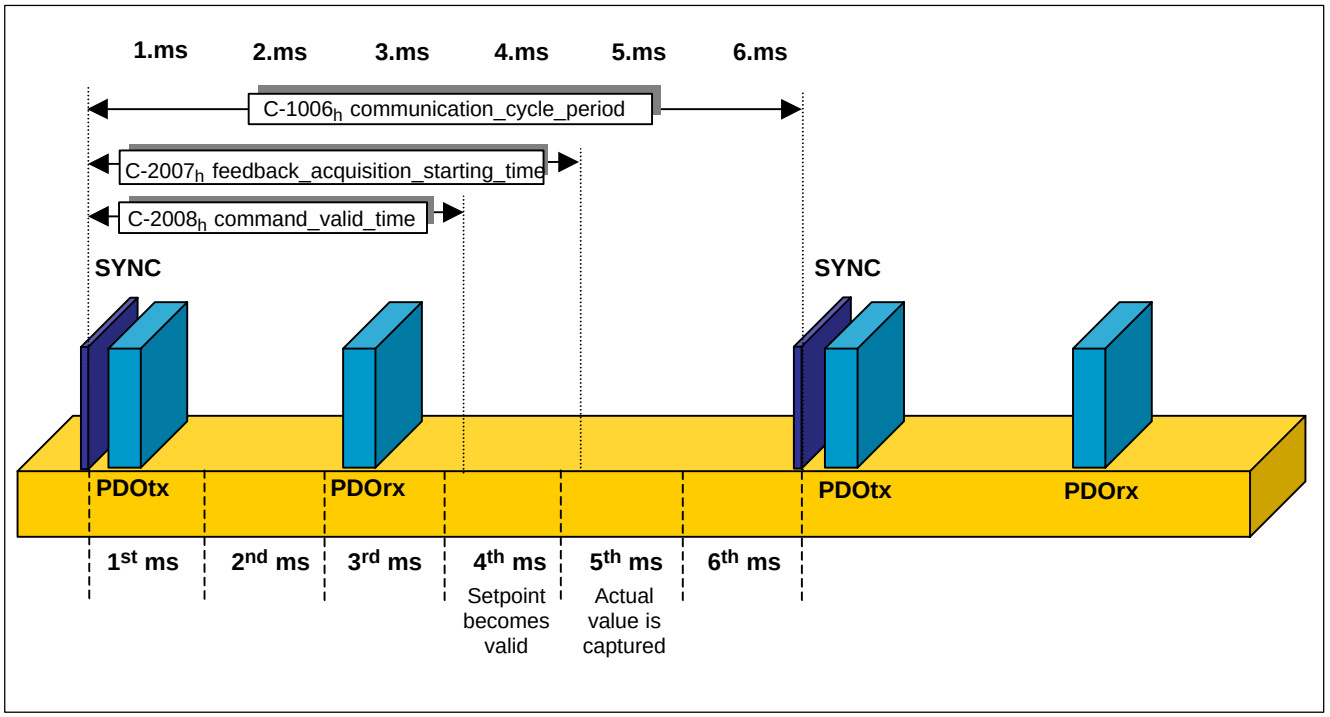
- **Example:**

  At a cycle time of 6 ms (C-1006 = 6000), C-2007 should be 5. The actual values are then measured in the 5th millisecond after the last SYNC telegram.

The setpoint valid time should be as close as possible after receipt of the PDOrx telegram. However, it is necessary to make sure that the telegrams have been received by the drive prior to this point in time.

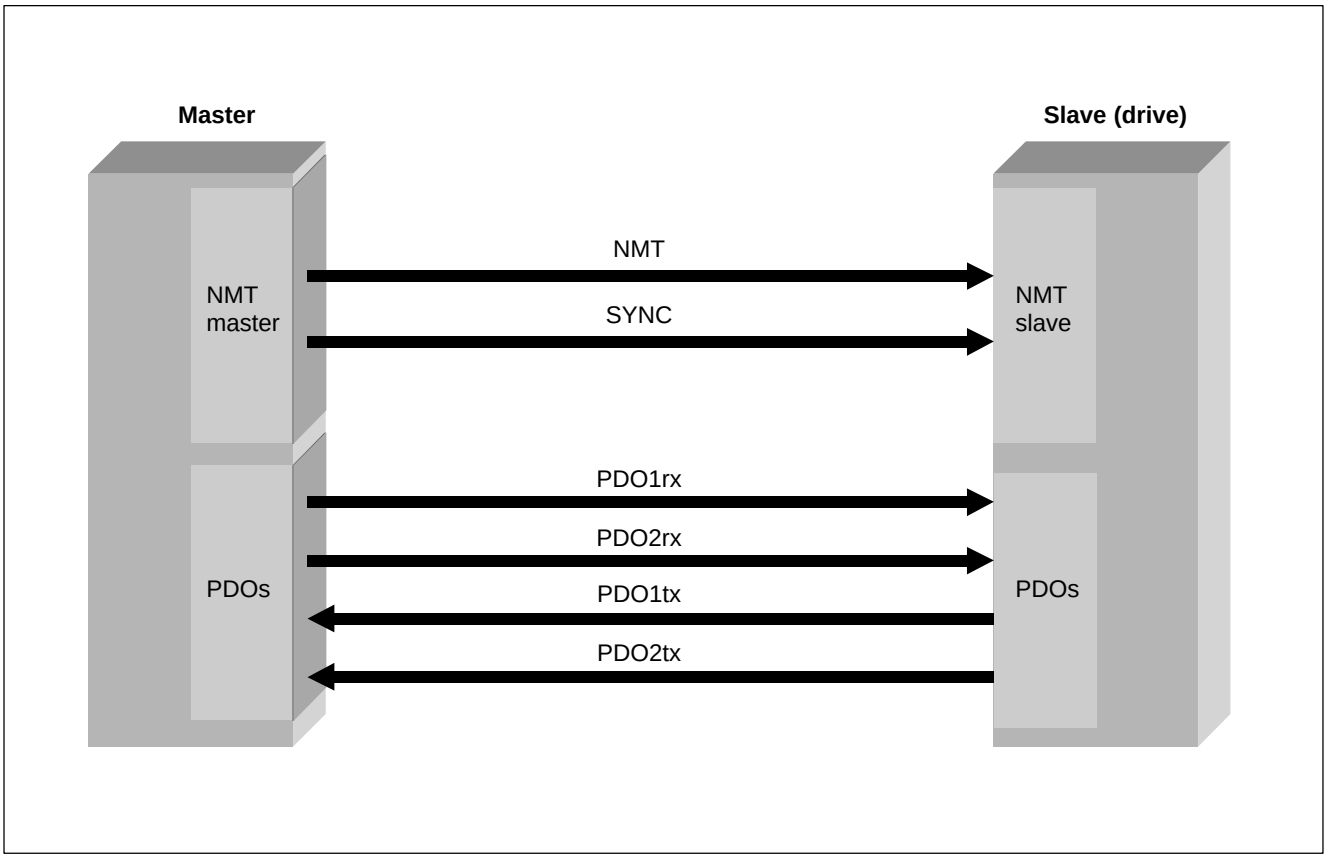
- **Example:**

  If the master sends the PDOrx telegrams in the 3rd millisecond after the SYNC, C-2008 may be 4. The drive then accesses the new setpoint in the 4th millisecond after the SYNC.

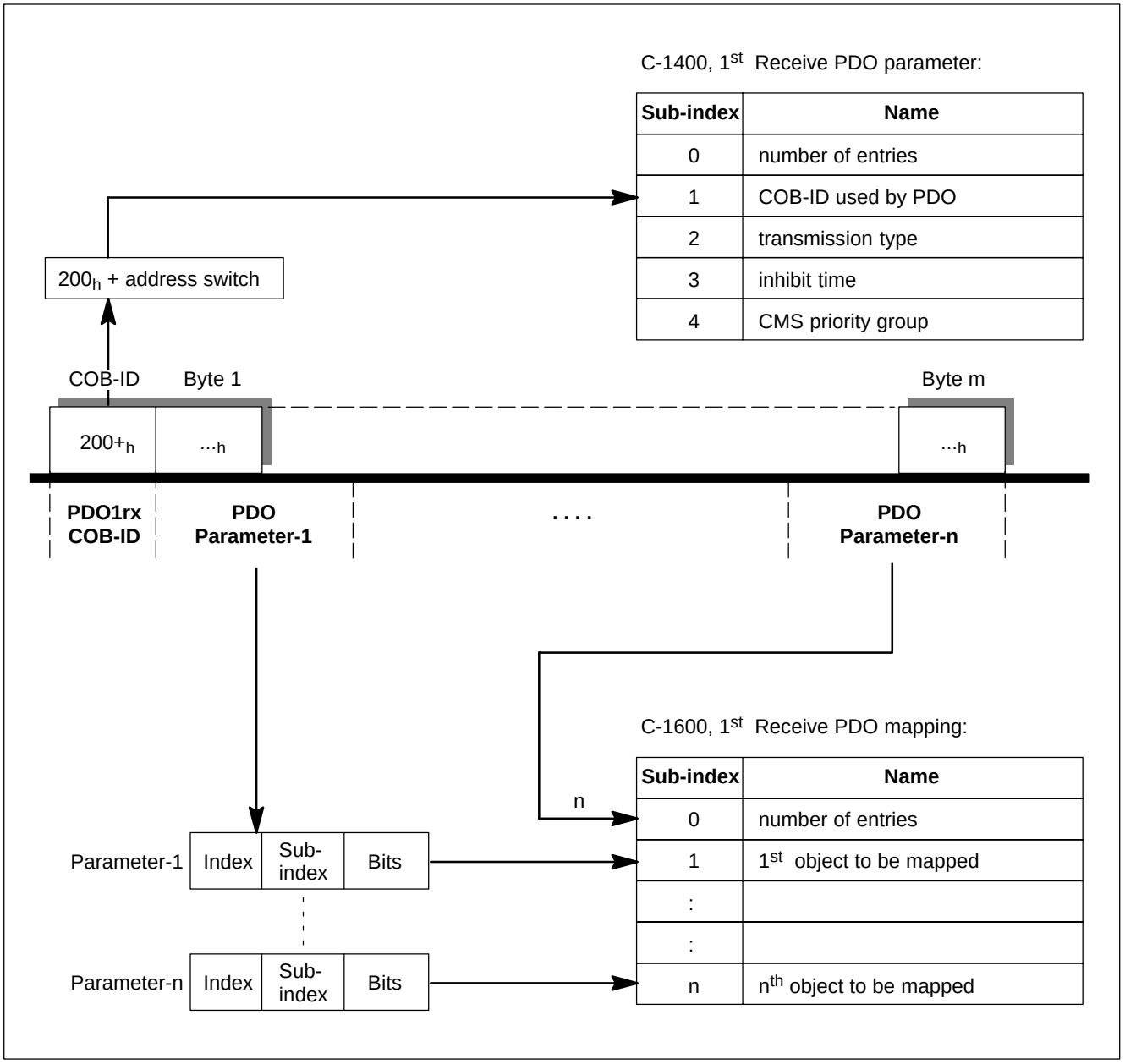


4.4 Process Data (PDO)

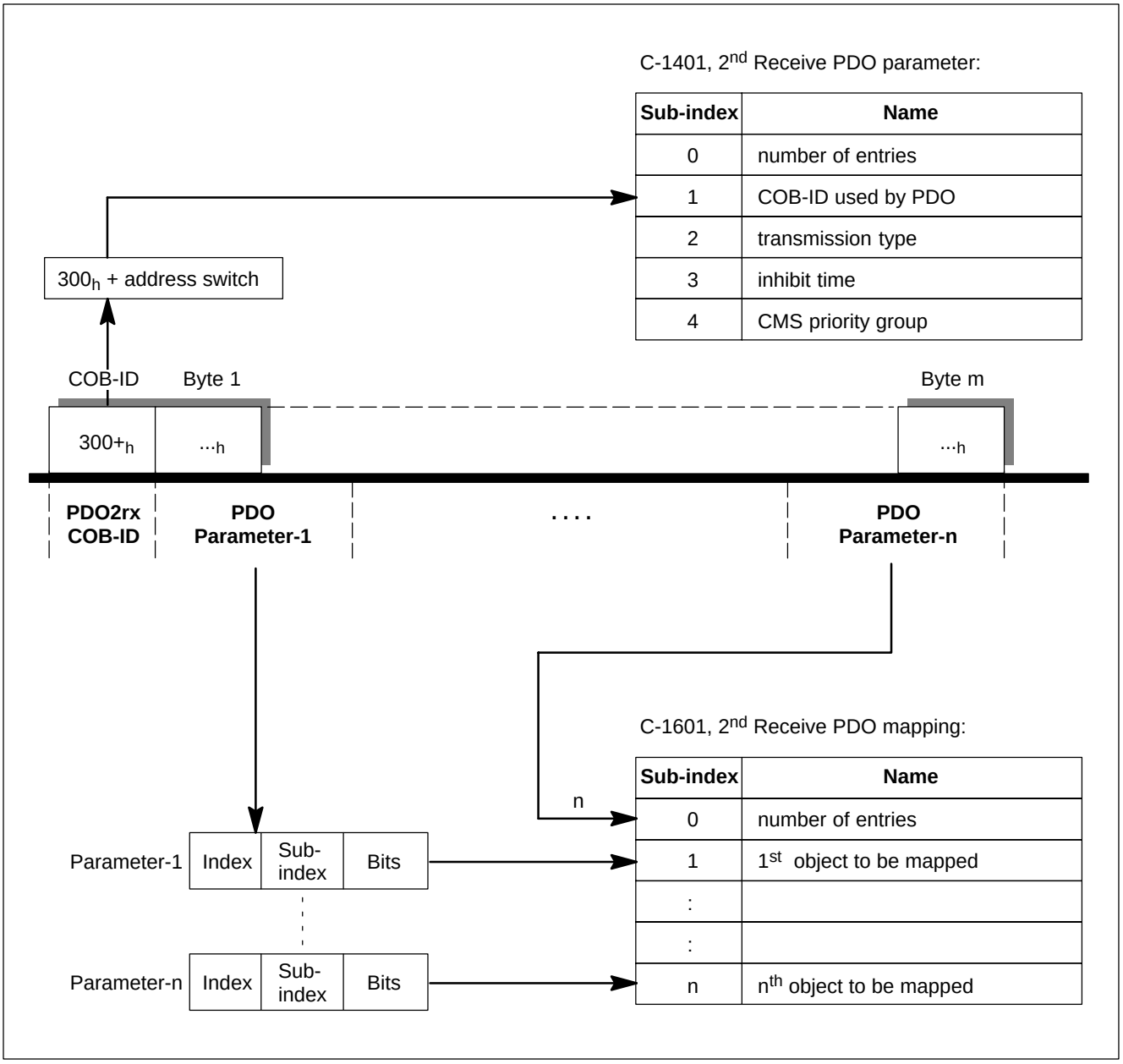
PDO Message



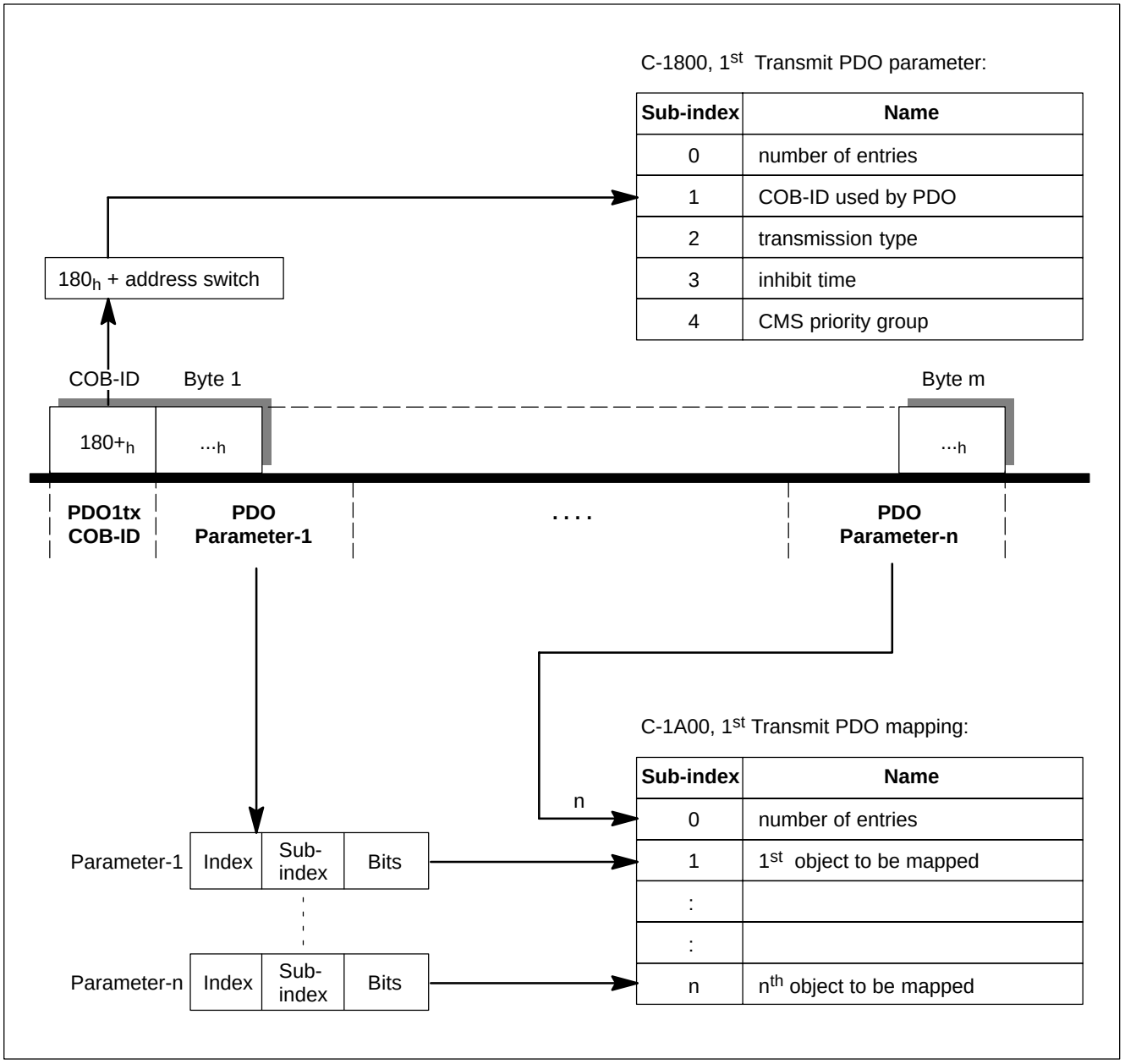
4.4.1 PDO1rx (Master to Slave)



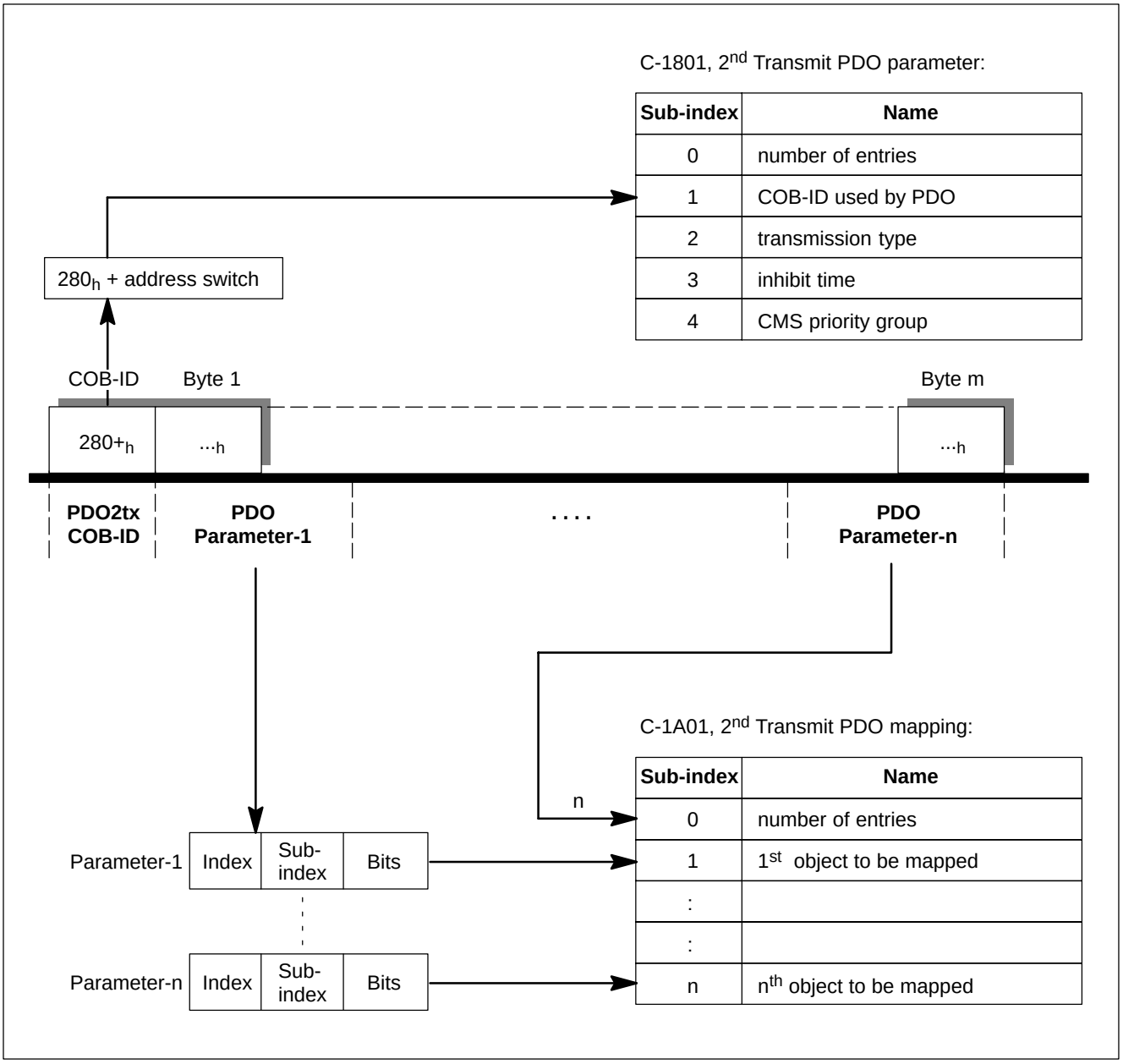
4.4.2 PDO2rx (Master to Slave)



4.4.3 PDO1tx (Slave to Master)

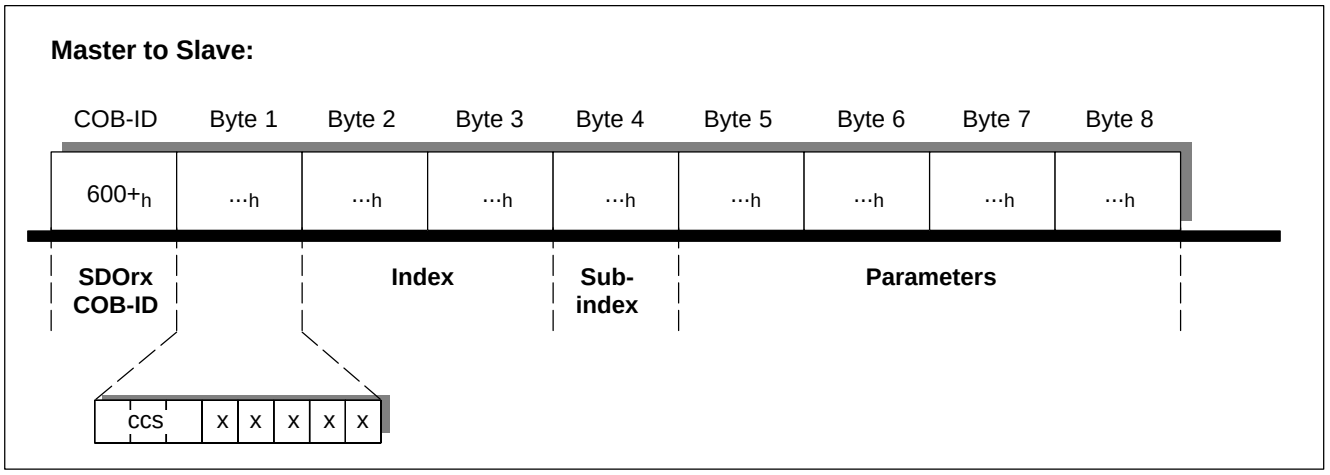


4.4.4 PDO2tx (Slave to Master)

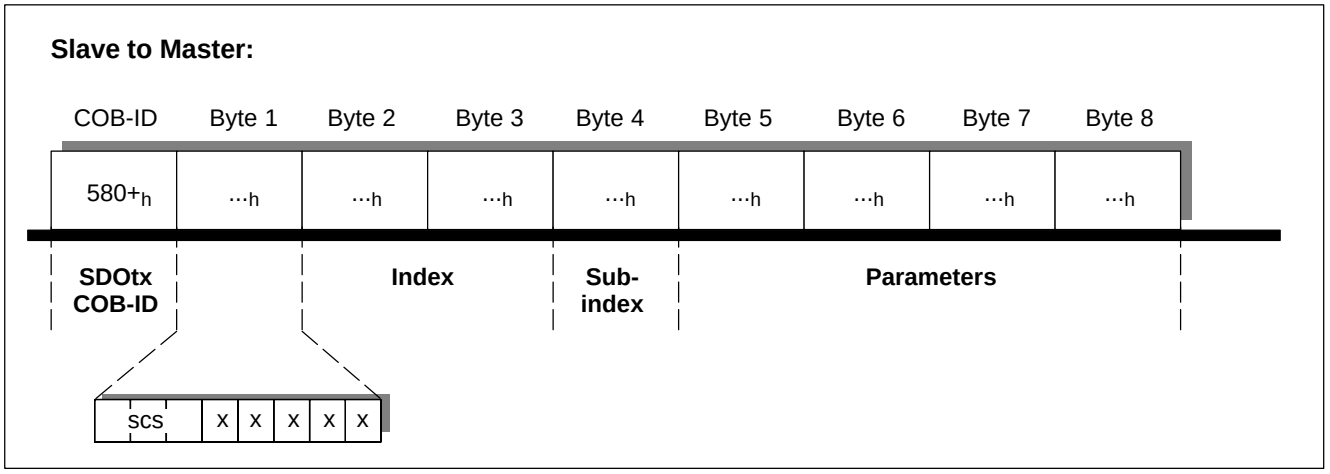


4.5 Service Data (SDO)

SDOrx



SDOtx



Coding

Master ccs (bin)	Slave scs (bin)	Protocol	Description
010xxxxx		SDO Upload Protocol request	DS-301
	010xxxxx	SDO Upload Protocol response	
011xxxxx		Upload SDO Segment Protocol request	DS-301
	000xxxxx	Upload SDO Segment Protocol response	
001xxxxx		SDO Download Protocol request	DS-301
	011xxxxx	SDO Download Protocol response	
000xxxxx		Download SDO Segment Protocol request	DS-301
	001xxxxx	Download SDO Segment Protocol response	
	1xxxxxxx	Upload/Download Error Error code: Byte 4 (LSB) ...7 (MSB)	

4.6 Emergency Messages (EMCY)

EMCY Messages

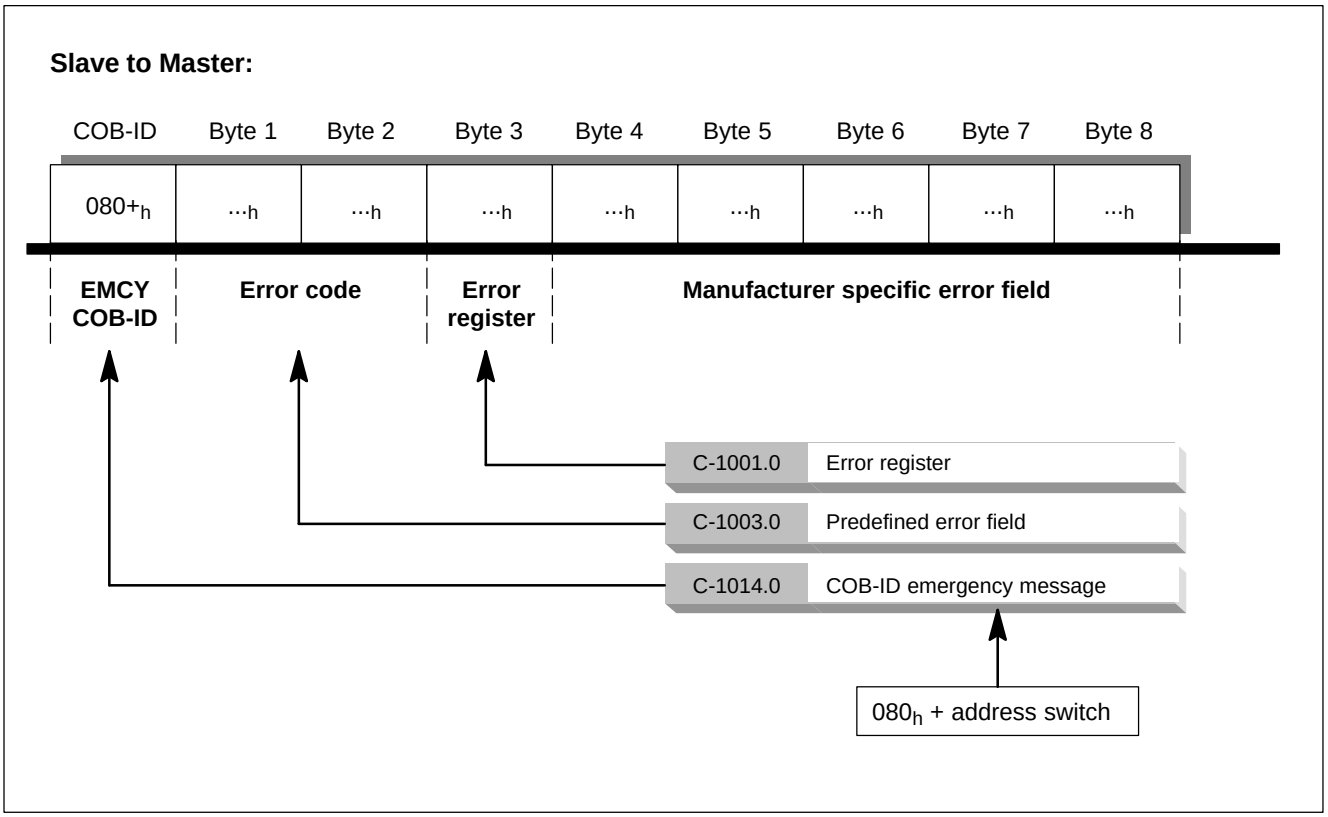
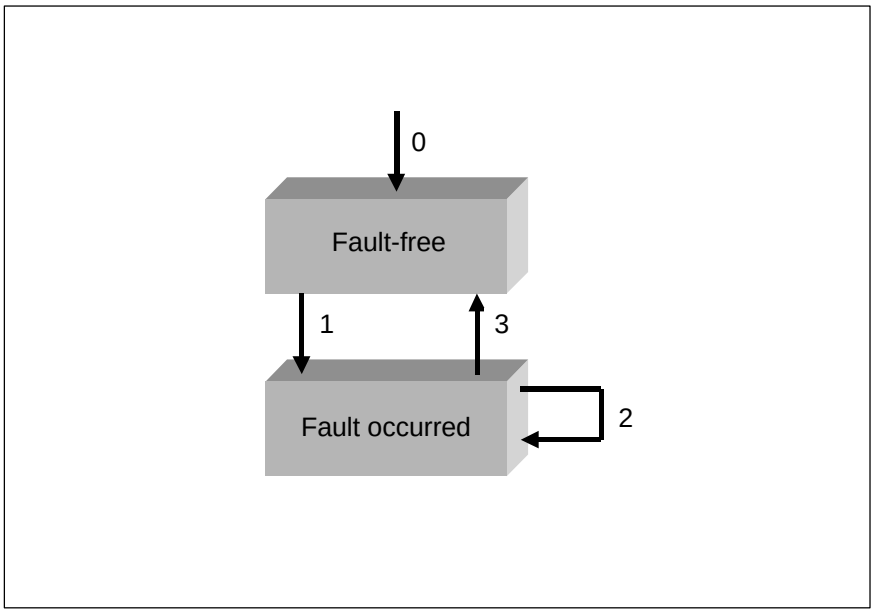


Diagram – Transition to Fault State



0	Following initialisation, the drive transitions to a fault-free state if no fault is detected. No emergency message is transmitted.
1	The drive recognised an internal fault in the first three bytes of the emergency message (error code and error register). The drive transitions to fault state. An emergency object with the corresponding error code and error register is transmitted. The error code is entered at the location of object C-1003 (”predefined error field”).
2	A new fault has occurred in the drive. The drive stays in fault state and transmits an emergency object with the corresponding error code. The new error code is entered above in the error code field. The respective error code is cleared from the field upon elimination of a fault. It must be ensured that the error codes are sorted in accordance with time (the oldest fault has the highest sub-index).
3	All faults have been eliminated. The drive transitions to fault-free state and transmits an emergency object with the error code ”reset error / no error”.

Notes:

5 Drive Data

5.1 Overview

Objects 6500h to 65FFh serve as database for drive parameters.
The manufacturer specific data records (RECORD) 6410_h and 6510_h contain detailed information on the drive components used. They are read automatically from the electronic nameplates.

5.2 Object Description

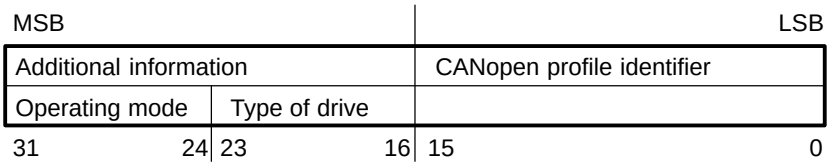
Objects defined in this chapter

Index	Code	Name	Type	Attr.
1000_h		device_type		
$20C8_h$	VAR	amplifier_temperature_warning	Unsigned16	rw
$20C9_h$	VAR	motor_temperature_warning	Unsigned16	rw
$20CA_h$	VAR	amplifier_shutdown_temperature	Unsigned16	ro
$20CB_h$	VAR	motor_shutdown_temperature	Unsigned16	ro
$20CC_h$	VAR	drive_on_delay_time	Unsigned16	rw
$20CD_h$	VAR	drive_off_delay_time	Unsigned16	rw
4001_h	VAR	switching_frequency _of_power_output_stage	Unsigned16	rw
4006_h	VAR	type_of_motor_encoder	Unsigned16	rw
4011_h	VAR	sync_enable	Unsigned16	rw
$41F9_h$	VAR	short_circuit_break_enable _delay_time	Unsigned16	rw
$424E_h$	VAR	drive_off_safety_delay_time	Unsigned16	rw
$47D0_h$	ARRAY	inport_configuration_list	Unsigned32	rw
$47D1_h$	VAR	inport_signal_control_word	Unsigned16	rw
$47D2_h$	ARRAY	outport_configuration_list	Unsigned32	rw
$47D3_h$	VAR	outport_signal_status_word	Unsigned16	rw
6410_h	RECORD	motor_data	RECORD	ro
6502_h	VAR	supported_drive_modes	Unsigned32	ro
6510_h	RECORD	drive_data	–	rw

C-1000$_h$ device_type

This object describes the type of drive and its functionality.
It contains information on the fulfilled CANopen profile in addition to the type of drive.

Data description



CANopen profile identifier	402	
Additional information		
Drive type bit encoded: Bits 16...23	Bit 16 = 1 Bit 17 = 1 Bit 18 = 1 Bit 23 = 1	Frequency inverter Servo drive Stepper motor I/O module (multiple device modules only)
Manufacturer specific: Bits 24...31	0	

C-1001$_h$ Error Register

The device specific bit in the status word is used by the CANopen Device Profile for Drives and Motion Control. The error code can be read from the predefined error field at object 1003h and has to be compatible with the device profiles for drives.

C-20C8$_h$ amplifier_temperature_warning

Temperature warning for inverter monitoring.

Index	**C-20C8$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	°C	
Value range	0...165	
Default value	75	

C-20C9$_h$ motor_temperature_warning

Temperature warning for motor monitoring.

Index	C-20C9$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	°C	
Value range	0...165	
Default value	145	

C-20CA$_h$ amplifier_shutdown_temperature

Switch-off temperature for inverter monitoring.

Index	C-20CA$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Units	°C	
Value range	–	
Default value	145	

C-20CB$_h$ motor_shutdown_temperature

Switch-off temperature for motor monitoring.

Index	C-20CB$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	°C	
Value range	0...165	
Default value	165	

C-20CC$_h$ drive_on_delay_time

After setting the signal ”Enable Operation” (bit 3 in the control word), the drive does not follow the setpoint inputs until the waiting time specified here has elapsed.
The torque becomes effective immediately, regardless of this waiting time.

Index	C-20CC$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...65535	
Default value	0	

CAUTION
Extreme wear of the holding brake!
The waiting time C-20CC$_h$ may not elapse before the brake has been completely released!

C-20CD$_h$ drive_off_delay_time

After the signal ”Enable Operation” (bit 3 in the control word) has been cleared, the torque is maintained in the drive until the waiting time specified here has elapsed.

Index	**C-20CD$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...65535	
Default value	0	

CAUTION
Extreme wear of the holding brake!
The holding brake is not an operation brake and may only be applied when the axle stands still.
To avoid damaging the holding brake, the motor has to reach C-20CD$_h$ n = 0 before the waiting time has elapsed.

Therefore take into account

- the required braking time for the drive to reach n = 0, when shutting down with ramp including ramp time, and the
- switch-off time of the brake until it is blocked (see corresponding Motor Manual).

☞ **See also C-424E$_h$.**

C-4001$_h$ switching_frequency_of_power_output_stage

This object defines the switching frequency of the power output stage.
High switching frequencies result in lower noise but also in lower output currents.
Thus, the switching frequency must be selected according to the motor-module assignment, i.e. the drive configuration.

Index	**C-4001$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	2000...8000	
Default value	8000	

C-4006$_h$ type_of_motor_encoder

Motors are available with single-turn (STG) or multi-turn (MTG) absolute encoders. While the STG allows for determination of the absolute motor position with respect to 1 revolution only, the MTG can signal its absolute position with respect to 4096 revolutions.
Both designs can be operated as incremental or absolute encoders.

Index	**C-4006$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = incremental encoder 1 = absolute encoder	
Default value	0	

C-4011$_h$ sync_enable

Using this object, it is possible to influence the synchronisation of the cycle frequency of the connected drives.

Index	**C-4011$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = inverter is synchronised to SYNC signal received from another inverter 1 = inverter transmits SYNC signal to all connected inverters 2 = SYNC signal is transmitted to all connected inverters via input IN4 3 = CAN controller sends SYNC signal to all connected inverters	
Default value	3	

C-41F9$_h$ short_circuit_break_enable_delay_time

For applications with a plug braking contactor this parameter must be set to a value > 0.
The torque will not be active, when enable operation has been set, unless the time set in this parameter has elapsed. Afterwards, the drive will follow the setpoint input.

Index	**C-41F9$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...1000	
Default value	0	

C-424E$_h$ drive_off_safety_delay_time

Specifies the maximum waiting time for switching off the output stage following ”Drive Off”.

Possible input:

- 0:
  “drive_off_safety_delay_time” is switched off. In this case, the output stage is only switched off at n=0.
- C-20CD$_h$ < C-424E$_h$:
  “drive_off_safety_delay_time“ is switched on.
  The output stage is switched off, even if n=0 has not been reached (e.g. in case of incorrect encoder displacement and overspeeding motor).
- C-424E$_h$ < C-20CD$_h$:
  Output stage is switched off when waiting time C-20CD$_h$ has elapsed.

Index	**C-424E$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...1000	
Default value	0	

C-47D0$_h$ inport_configuration_list

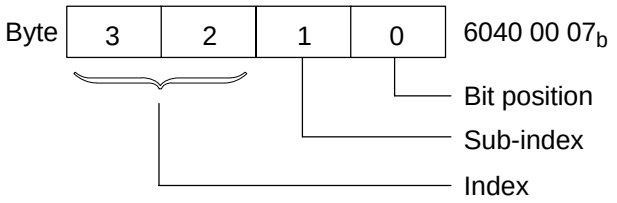
This object specifies the assignment of the digital inputs IN1...IN4 on terminal block X06. Thus it is possible to assign different functionalities to the inputs.

★ Enter the object to be assigned to the input indicating index, sub-index and bit position into the corresponding sub-index of C-47D0. Each sub-index is assigned to a specified port. The value 0 is for unassigned inputs.

Example:

Reset error messages via IN2 (X06.6) (control word C-6040, bit 7).

Sub-index 02 is assigned: C-47D0.02 = 6040 00 07$_b$



Index	C-47D0$_h$
Object code	ARRAY
Number of elements	4
Data type	Unsigned32

digital_input_1 (IN1)

Sub-index	01$_h$	
Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Value range	–	
Default value	0	

digital_input_2 (IN2)

Sub-index	02$_h$	
Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Value range	–	
Default value	0	

digital_input_3 (IN3)

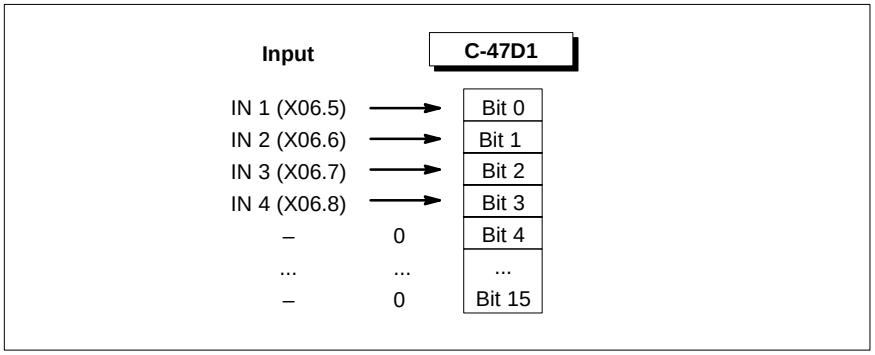
Sub-index	03$_h$	
Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Value range	–	
Default value	0	

digital_input_4 (IN4)

Sub-index	04$_h$	
Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Value range	–	
Default value	0	

C-47D1$_h$ inport_signal_control_word

This object maps the signal statuses of the digital inputs IN1 to IN4 to the first 4 bits.
A voltage level between +15 VDC and +30 VDC at the inputs will be interpreted as ”logic 1” by the drive, and the corresponding bit will be set.



Index	C-47D1$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	–	
Default value	–	

C-47D2$_h$ outport_configuration_list

This object specifies the assignment of the relay output Out1 on terminal block X34. Thus it is possible to output the logical status of the object.

☞ **The digital outputs Out2...Out4 are always preassigned for:**

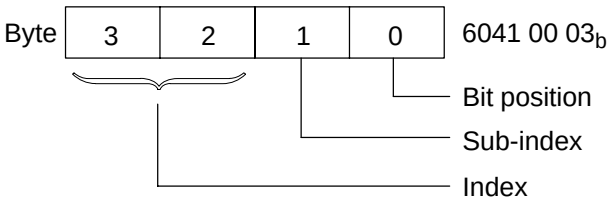
- **Overtemperature warning Out2**
- **Control of holding brake Out3**
- **Control of plug braking contactor Out4**

★ Enter the object to be assigned to the output indicating index, sub-index and bit position into the corresponding sub-index of C-47D2. The value 0 is for the unassigned output.

Example:

Assign group error message in the status word (C-6041, bit3) to relay output Out1 (X34.4,5,6).

Sub-index 01 is assigned: C-47D2.01 = 6041 00 03$_b$



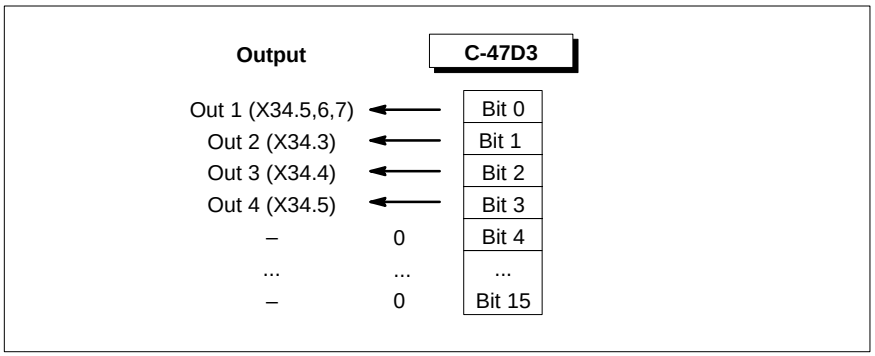
Index	**C-47D2$_h$**
Object code	ARRAY
Number of elements	1
Data type	Unsigned32

digital_output_1 (Out1)

Sub-index	**01$_h$**	
Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Value range	–	
Default value	0	

C-47D3$_h$ outport_signal_status_word

The first 4 bits of this object connect the relay output Out1 and set the digital outputs Out2...Out4 to +24 VDC. Thus the objects selected under C-47D2 and the predetermined objects penetrate to the outputs.



Index	C-47D3$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	–	
Default value	–	

C-6410$_h$ motor_data

This object contains comprehensive information about the connected motor. All data are read automatically from the electronic nameplate.

Index	C-6410$_h$
Object code	RECORD
Number of elements	21

motor_ID

Sub-index	01$_h$	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	visible string	

motor_type

Sub-index	**02_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	Bit 0: 1 = servo motor Bit 1: 1 = spindle motor Bit 2: 1 = standard motor	
Data type	Unsigned16	

option

Sub-index	**03_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	Bit 0: 1 = servo motor Bit 1: 1 = spindle motor Bit 2: 1 = standard motor	
Data type	visible string	

encoder_type

Sub-index	**04_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	Bit 0: 0 = single-turn encoder 1 = multi-turn encoder Bit 1: 0 = degree of protection IP00 1 = degree of protection IP65 Bit 2: 1 = gearwheel encoder Bit 3: – Bit 4: 1 = variable encoder pulses Bit 5: 1 = resolver	
Data type	Unsigned16	

encoder_pulses

Sub-index	**05_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	number of pulses/rev	
Data type	Unsigned16	

encoder_offset

Sub-index	**06_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	increments	
Data type	Unsigned16	

version_no

Sub-index	**07_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

rated_velocity

Rated speed of motor.

Sub-index	**08_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	rpm	
Data type	Unsigned16	

max_velocity

Maximum speed of motor.

Sub-index	**08_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	rpm	
Data type	Unsigned16	

rated_current

Rated current of motor.

Sub-index	**10_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

standstill_current

Standstill current of motor.

Sub-index	**11_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

max_current

Maximum current of motor.

Sub-index	**12_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

rated_torque

Rate torque of motor.

Sub-index	**13_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 Nm	
Data type	Unsigned32	

standstill_torque

Standstill torque of motor.

Sub-index	**14_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 Nm	
Data type	Unsigned32	

max_torque

Maximum torque of motor.

Sub-index	**15_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 Nm	
Data type	Unsigned32	

rated_voltage

Rated voltage of motor.

Sub-index	**16_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[V]	
Data type	Unsigned16	

pair_of_poles

Number of pair of poles of motor.

Sub-index	**17_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

brake

Motor with/without holding brake

Sub-index	**18_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0 = without holding brake 1 = with holding brake	
Data type	Unsigned16	

software_index

Sub-index	**19_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

emk_constant

Sub-index	**20_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 Vmin	
Data type	Unsigned32	

C-6502$_h$ supported_drive_modes

This object shows which modes of operation are supported by the drive.

Index	**C-6502$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: –
Access	ro	
PDO mapping	possible	
Units	–	
Value range	0..(2^{32} –1)	
Default value	–	

Data description

Bit number	**Description**
0	Profile Position Mode
1	Velocity Mode
2	Profile Velocity Mode
3	Profile Torque Mode
4	reserved
5	Homing Mode
6	Interpolated Position Mode
7	reserved
8	reserved
9	reserved
10...15	reserved
16	Motion Control Mode
17...31	reserved

C-6510$_h$ drive_data

This object contains comprehensive information about the drive inverter. All data are read automatically from the electronic nameplate.

Index	**C-6510$_h$**
Object code	RECORD
Number of elements	21

amplifier_ID

Sub-index	**01$_h$**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	visible string	

version_no

Sub-index	**02$_h$**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

version_index

Sub-index	**03$_h$**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

board_no

Identifier of the computer board in the drive inverter.

Sub-index	**04$_h$**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned32	

board_index

Output status of the computer board in the drive inverter.

Sub-index	**05_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	–	
Data type	Unsigned16	

rated_current

Rated current of the power output stage.

Sub-index	**06_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

max_permissible_overcurrent

Maximum permissible current at maximum overload.

Sub-index	**07_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

elco_capacity

DC-link capacity contained in inverter

Sub-index	**08_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[µF]	
Data type	Unsigned32	

max_rated_current

Maximum permissible rated current at maximum continuous load.

Sub-index	**09_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

max_peak_current

Maximum current at maximum continuous load.

Sub-index	**10_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

max_heat_sink_temperature

Maximum permissible heat sink temperature.

Sub-index	**11_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[°C]	
Data type	Unsigned16	

max_switching_frequency

Maximum permissible switching frequency.

Sub-index	**11_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[Hz]	
Data type	Unsigned16	

dc_link_rated_voltage

Rated voltage of DC-link.

Sub-index	**11_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[Hz]	
Data type	Unsigned16	

max_permissible_overcurrent_level

Shutdown limit for maximum permissible current at maximum overload.

Sub-index	**14_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

drive_max_current

Module maximum current.

Sub-index	**15_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

drive_max_current_effective

Module maximum current, rms.

Sub-index	**16_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

drive_rated_current_effective

Module rated current, rms.

Sub-index	**17_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	0.001 A	
Data type	Unsigned32	

switching_frequency

Active switching frequency.

Sub-index	**18_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[Hz]	
Data type	Unsigned16	

heat_sink_shutdown_temperature

Shutdown temperature of drive inverter.

Sub-index	**19_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[°C]	
Data type	Unsigned16	

heat_sink_temperature_warning

Warning temperature of drive inverter.

Sub-index	**20_h**	
Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Value range	[°C]	
Data type	Unsigned16	

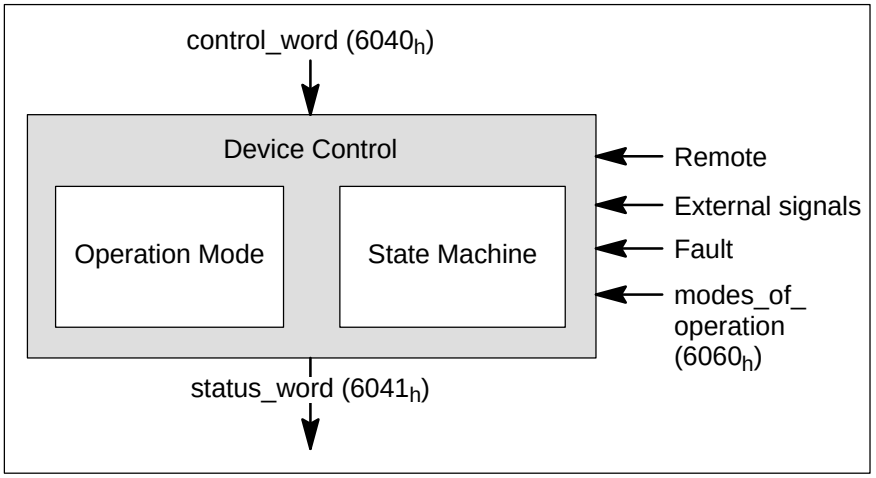
Notes:

6 Device Control

6.1 Structure

The device control function block controls all functions of the drive, i.e. drive functionality and power section. It is divided into:

- State Machine and
- Modes of Operation Function.

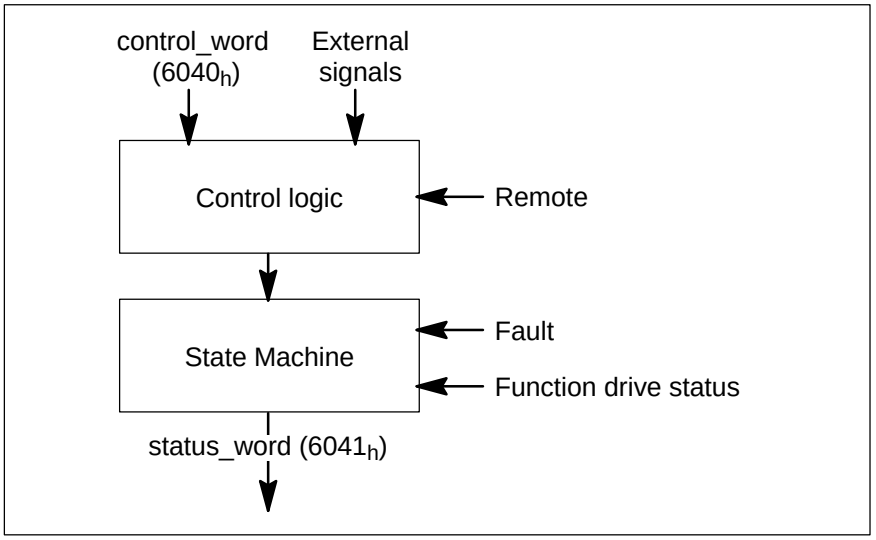


The state of the drive is controlled by the **control word**.
The **status word** indicates the state of the drive.

The **State Machine** is influenced by:

- the control word
- external signals and
- internal signals such as faults and modes_of_operation.

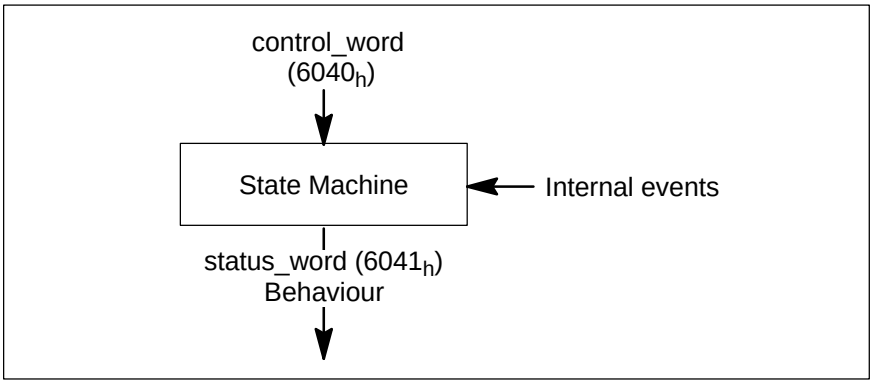
In **Remote Mode** the device is controlled directly from the CAN network by Process Data Objects (PDOs) and Service Data Objects (SDOs).



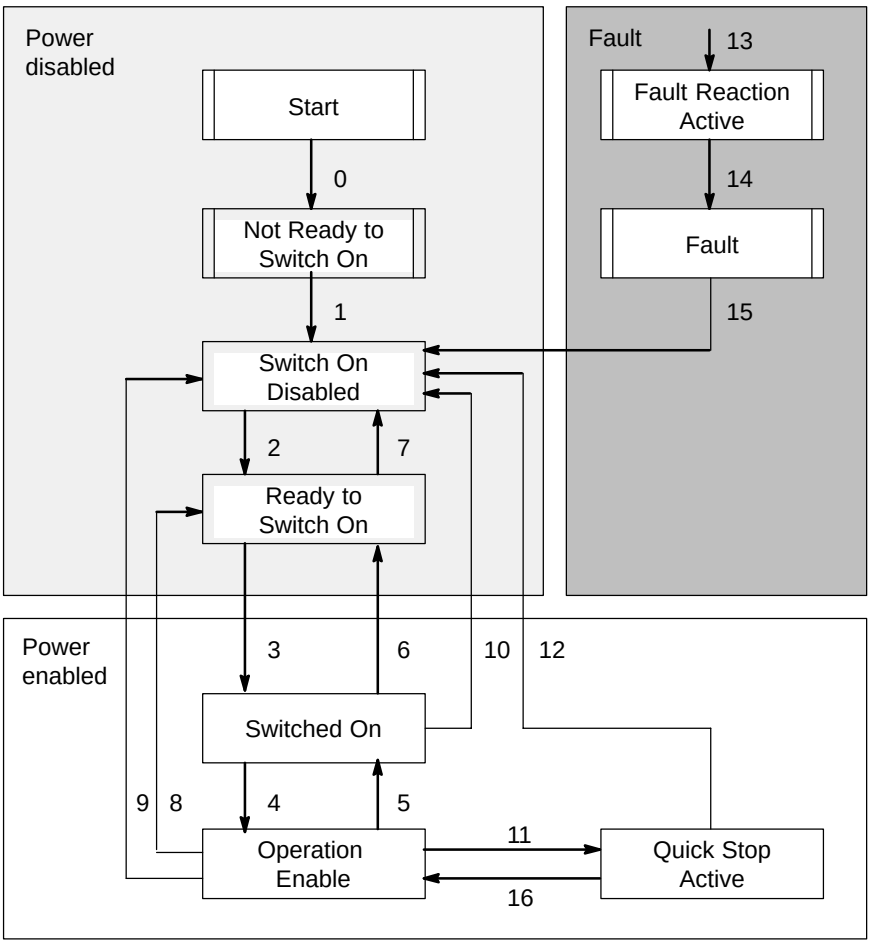
6.2 State Machine

The State Machine describes the status and the possible control sequence of the drive. Each state represents a special internal or external behaviour. The state of the drive also determines which commands are accepted. E.g. it is only possible to start a point-to-point move when the drive is in state OPERATION ENABLED.

The state of the State Machine is changed accordingly, using the control word and/or internal events. The current state can be read using the status word.



The following state diagram describes the State Machine with regard to the control of the power section, depending on user commands and internal drive faults.



6.2.1 States

The following states of the drive are possible:

- **Not Ready to Switch On:**
  24 V power supply present, no release FG (X06.3 = 0).
  The drive is being initialised or is running self test.
  A motor brake, if present, has been applied.
  The operation is disabled.

- **Switch On Disabled:**
  Drive initialisation is complete.
  Release FG has been given (X06.3 = 24V).
  The drive parameters have been set.
  Drive parameters may be changed.
  The power supply is disabled.
  The operation is disabled.

- **Ready to Switch On:**
  Power has been applied.
  Drive parameters may be changed.
  The operation is disabled.

- **Switched On:**
  Power has been applied.
  The output stage is ready.
  Drive parameters may be changed.
  The operation is disabled.

- **Operation Enabled:**
  Normal operation, no faults have been detected.
  The operation is enabled and the motor has torque.
  Drive parameters may be changed.

- **Quick Stop Active:**
  Quick Stop is executed.
  Drive parameters may be changed.
  The operation is enabled and the motor has torque.

☞ **If the object quick_stop_option_code (C-605A$_h$) = 5 (stay in the quick stop mode), you can only return to normal operation with the command 'Enable Operation'.**

- **Fault Reaction Active:**
  A non-fatal fault has occurred in the drive.
  Quick Stop is executed.
  Drive parameters may be changed.
  The operation is enabled and the motor has torque.

- **Fault:**
  A fault has occurred in the drive.
  Drive parameters may be changed.
  The operation is disabled.

☞ **When the operation is disabled, no energy is supplied to the motor. It is only when the operation is enabled that the motor is supplied with energy and the reference values (torque, speed, position) are processed.**

6.2.2 State Transitions

State transitions are caused by internal events in the drive or via the control word.

State Transition 0 **Startup→ Not Ready to Switch On**

Event: RESET
Action: The drive self-tests and initialises.

State Transition 1 **Not Ready to Switch On → Switch On Disabled**

Event: Release FG is given (X06.3 = 24 V) after the drive has initialised and self-tested successfully.
Action: Communication and process data monitoring are active.

State Transition 2 **Switch On Disabled → Ready to Switch On**

Event: Shutdown command received via control word.
Action: None

State Transition 3 **Ready to Switch On → Switched On**

Event: 'Switch On' command received via control word.
Action: The power section is activated.

State Transition 4 **Switched On → Operation Enabled**

Event: 'Enable Operation' command received via control word.
Action: The operation is enabled.

State Transition 5 **Operation Enabled → Switched On**

Event: 'Disable Operation' command received via control word.
Action: The operation is disabled.

State Transition 6 **Switched On → Ready to Switch On**

Event: Shutdown command received via control word.
Action: The power section is switched off.

State Transition 7 **Ready to Switch On → Switch On Disabled**

Event: 'Quick Stop' command received via control word.
Action: None.

State Transition 8 **Operation Enabled → Ready to Switch On**

Event: Shutdown command received via control word.
Action: The power section is switched off immediately, and the motor runs down torque-free if unbraked.

State Transition 9 **Operation Enabled → Switch On Disabled**

Event: 'Disable Voltage' command received via control word.
Action: The power section is switched off immediately, and the motor runs down torque-free if unbraked.

State Transition 10 **Switched On → Switch On Disabled**

Event: 'Disable Voltage' or 'Quick Stop' command received via control word.
Action: The power section is switched off immediately, and the motor runs down torque-free if unbraked.

State Transition 11 **Operation Enabled → Quick Stop Active**

Event: 'Quick Stop' command received via control word.
Action: Quick Stop is executed.

State Transition 12 **Quick Stop Active → Switch On Disabled**

Event: 'Disable Voltage' command received via control word or the quick stop is completed (only possible if the object quick-stop-option-code is different 5).
Action: The power section is switched off.

State Transition 13 **all states → Fault Reaction Active**

Event: A fault has occurred in the drive.
Action: The drive executes appropriate fault reaction.

☞ **When a fault occurs, there is a transition of the drive to the state 'Fault' via the state 'Fault Reaction Active'. This state can only be left with the command 'Fault Reset' if no fault exists**

State Transition 14 **Fault Reaction Active → Fault**

Event: The fault reaction is completed.
Action: The operation is disabled. The power section is switched off.

State Transition 15 **Fault → Switch On Disabled**

Event: 'Fault Reset' command received via control word.
Action: A reset of the fault condition is carried out if no fault exists currently on the drive. After leaving the 'Fault' state the bit 'Fault Reset' of the control word has to be cleared.

State Transition 16 **Quick Stop Active → Operation Enable**

Event: 'Enable Operation' command received via control word. Only possible if the object quick-stop-option-code is 5, 6, 7 or 8 (see page 6–13).
Action: The operation is enabled.

6.3 Object Description

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
6040_h	VAR	control_word	Unsigned16	rw
6041_h	VAR	status_word	Unsigned16	rw
$605B_h$	VAR	shutdown_option_code	Integer16	rw
$605C_h$	VAR	disable_operation_option_code	Integer16	rw
$605A_h$	VAR	quick_stop_option_code	Integer16	rw
$605D_h$	VAR	stop_option_code	Integer16	rw
$605E_h$	VAR	fault_reaction_option_code	Integer16	rw
6060_h	VAR	modes_of_operation	Integer8	wo
6061_h	VAR	modes_of_operation_display	Integer8	ro

C-6040_h control_word

The bits set in the control_word and the signal transitions of external signals result in the Device Control command. The control_word is always mapped into the first two bytes of the drive's command message. The bits of the control_word are defined as follows:

Index	C-6040_h
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: all	O: –
Access	rw	
PDO mapping	possible	
Units	–	
Value range	0...65535	
Default value	–	

Data description

MSB LSB

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

High byte Low byte

Bit	Name
0	Switch On
1	Disable Voltage
2	Quick Stop
3	Enable Operation
4	Operation Mode Specific
5	Operation Mode Specific
6	Operation Mode Specific
7	Reset Fault
8	Halt
9	Reserved
10	Reserved
11	Operation Mode Specific
12	Reserved
13	Reserved
14	Reserved
15	Reserved

Bit 0, 1, 2, 3, 7 **Device Control commands** are triggered by these bits in the control word:

Command/ bit of the control word	Bit 7 Reset Fault	Bit 3 Enable Operation	Bit 2 Quick Stop	Bit 1 Disable Voltage	Bit 0 Switch On	Transi-tions
Shutdown	0	X	1	1	0	2,6,8
Switch On	0	X	1	1	1	3
Disable Voltage	0	X	X	0	X	7,9,10,12
Quick Stop	0	X	0	1	X	7,10,11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4,16
Reset Fault		X	X	X	X	15

☞ **Bit 3 ”Enable Operation” acts time dependent on the objects:**

- **C-20CC (drive_on_delay_time)**
- **C-20CD (drive_off_delay_time)**
- **C-41F9 (short_circuit_break_delay time)**
- **C-424E (drive_off_safety_delay_time)**

(see Section 5.2).

Bit 4, 5, 6, 8, 11 Meaning depends on the mode of operation:

Bit	Operation Mode					
	Velocity Mode	**Profile Position Mode**	**Profile Velocity Mode**	**Profile Torque Mode**	**Homing Mode**	**Interpol. Position Mode**
4	–	new_set-point	reserved	reserved	Start Homing Mode	enable_ip_mode
5	–	–	reserved	reserved	reserved	reserved
6	–	0: absolute 1: relative	reserved	reserved	reserved	reserved
8	–	Halt*	–	–	–	–
11	–	Reset Interpolator	–	–	–	–

* 'Halt' interrupts the move of a drive, and waits for release to continue

Bit 9, 10, 12, 13, 14, 15 These bits are reserved, they have to be set to zero.

C-6041$_h$ status_word

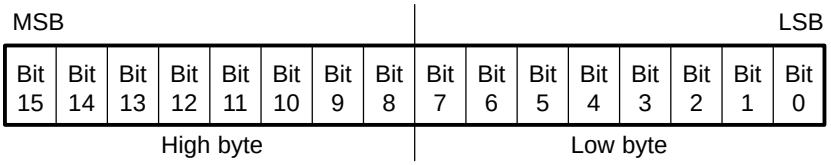
The status_word indicates the current status of the drive and is always mapped into the first two bytes of the actual message. The following bits are defined in the status_word.

Index	**C-6041$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: all	O: –
Access	ro	
PDO mapping	possible	
Units	–	
Value range	0...65535	
Default value	–	

Data description



Bit	**Name**
0	Ready to Switch On
1	Switched On
2	Operation Enabled
3	Fault
4	Voltage Disabled
5	Quick Stop
6	Switch On Disabled
7	Warning
8	Reserved
9	Reserved
10	Target Reached
11	Internal Limit Active
12	Operation Mode Specific
13	Operation Mode Specific
14	Operation Mode Specific
15	Reserved

Bit 0, 1, 2, 3, 5, 6 These bits indicate the state of the drive:

State	Bit 6 Switch On Disabled	Bit 5 Quick Stop	Bit 3 Fault	Bit 2 Oper-ation Enable	Bit 1 Switched On	Bit 0 Ready to Switch On
Not Ready to Switch On	0	X	0	0	0	0
Switch On Disabled	1	X	0	0	0	0
Ready to Switch On	0	1	0	0	0	1
Switched On	0	1	0	0	1	1
Operation En-abled	0	1	0	1	1	1
Fault	0	X	1	1	1	1
Fault Reaction Active	0	X	1	1	1	1
Quick Stop Active	0	0	0	1	1	1

Bits marked X are irrelevant for that state. Other bit combinations are not allowed.

Bit 4 voltage_disable

The Disable Voltage command can only be activated when the voltage_disabled bit is cleared to 0.

Bit 5 quick_stop

Feedback that the drive is reacting to the ’Quick Stop’ command. When bits 0, 1 and 2 of the status word are set to 1, the drive is capable of regenerating. The setting of the other bits indicates the status of the drive (e.g. the drive is performing a quick stop as result of a reaction to a non-fatal fault. The fault bit is set as well as bits 0, 1 and 2).

Bit 7 warning

Warning before a fault occurs. The status of the drive does not change. The cause of this warning may be found by reading the fault code parameter. The bit is set and reset by the drive.

Bit 8, 9, 15 These bits are reserved, they have to be set to zero.

Bit 10 target_reached

If bit 10 is set by the drive, then a setpoint has been reached (speed or position depending on the modes_of_operation). The change of a target value by software alters this bit.
If quickstop_option_code is 5, 6, 7 or 8, this bit must be set, when the quick stop operation is finished and the drive is halted.

Bit 11 **internal_limit_active**

This bit indicates, that an internal limitation is active (e.g. position_range_limit).

Bit 12, 13, 14 Meaning depends on the mode of operation:

Bit	Mode of operation					
	Velocity Mode	**Profile Position Mode**	**Profile Velocity Mode**	**Profile Torque Mode**	**Homing Mode**	**Interpol. Position Mode**
12	reserved	setpoint acknowl-edge	reserved	reserved	Homing attained	Ip mode active
13	reserved	following error	–	reserved	–	reserved
14	reserved	acknowl-edge inter-polator reset	reserved	reserved	reserved	reserved

☞ **All bits reflect the current state of the drive. No bits are latched.**

C-605B$_h$ shutdown_option_code

This object determines what action should be taken in case of transition 8 'Operation Enabled' → 'Ready to Switch On'.

Index	C-605B$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = disable drive function (motor runs down torque-free) 1 = slow down using slow down ramp according to C-605D$_h$	
Default value	0	

C-605C$_h$ disable_operation_option_code

This object determines what action should be taken in case of transition 5 'Operation Enabled' → 'Switched On'.

Index	C-605C$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = disable drive function (motor runs down torque-free) 1 = slow down using slow down ramp according to C-605D$_h$	
Default value	1	

C-605A$_h$ quick_stop_option_code

This object determines how the 'quick_stop' command is executed.

Index	C-605A$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = disable drive function (motor runs down torque-free) 1 = slow down according to slow down ramp 2 = slow down according to quick stop ramp 3 = slow down on the current limit 4 = – 5 = slow down according to slow down ramp and stay in Quick Stop 6 = slow down according to quick stop ramp and stay in Quick Stop 7 = slow down on the current limit and stay in Quick Stop	
Default value	2	

C-605D$_h$ stop_option_code

This object determines on which slow down ramp C-605B$_h$ and C-605C$_h$ are executed.

Index	C-605D$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = no slow down ramp (motor runs down torque-free) 1 = according to slow down ramp (C-6084$_h$ / C-6048$_h$) 2 = according to Quick-Stop ramp (C-6085$_h$) 3 = on the current limit (C-60C5h)	
Default value	1	

C-605E$_h$ fault_reaction_option_code

This object determines how the drive should react when a fault occurs.

Index	C-605E$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: all
Access	rw	
PDO mapping	no	
Units	–	
Value range	–32768..32767	
Value range	0 = disable drive function (motor runs down torque-free) 1 = slow down according to slow down ramp 2 = slow down according to quick stop ramp 3 = slow down on the current limit	
Default value	2	

C-6060$_h$ modes_of_operation

This object switches the mode of operation.

Index	C-6060$_h$
Object code	VAR
Data type	Integer8

Value description

Object class	M: all	O: –
Access	rw	
PDO mapping	possible	
Units	–	
Value range	–8 = Motion Control Mode 0 = – 1 = Profile Position Mode 2 = Velocity Mode 3 = Profile Velocity Mode 4 = Profile Torque Mode 5 = – 6 = Homing Mode 7 = Interpolated Position Mode	
Default value	–	

☞ **The modes_of_operation_display (C-6061$_h$) indicates the current mode of operation.**

C-6061$_h$ modes_of_operation_display

This object indicates the current mode of operation chosen with modes_of_operation (C-6060$_h$).

Index	**C-6061$_h$**
Object code	VAR
Data type	Integer8

Value description

Object class	M: all	O: –
Access	ro	
PDO mapping	possible	
Units	–	
Value range	–8 = Motion Control Mode 0 = – 1 = Profile Position Mode 2 = Velocity Mode 3 = Profile Velocity Mode 4 = Profile Torque Mode 5 = – 6 = Homing Mode 7 = Interpolated Position Mode	
Default value	–	

6.4 Drive Functions

6.4.1 Modes of Operation

The drive behaviour depends on the chosen mode of operation. Only one mode of operation can be active at a time, you can switch between the following modes of operation:

- Velocity Mode (simple 16-bit speed interface)
- Profile Velocity Mode
- Profile Torque Mode
- Homing Mode
- Profile Position Mode
- Interpolated Position Mode
- Motion Control Mode

With exception of the ’Homing Mode’ the listed modes of operation differ in the type of setpoint setting.

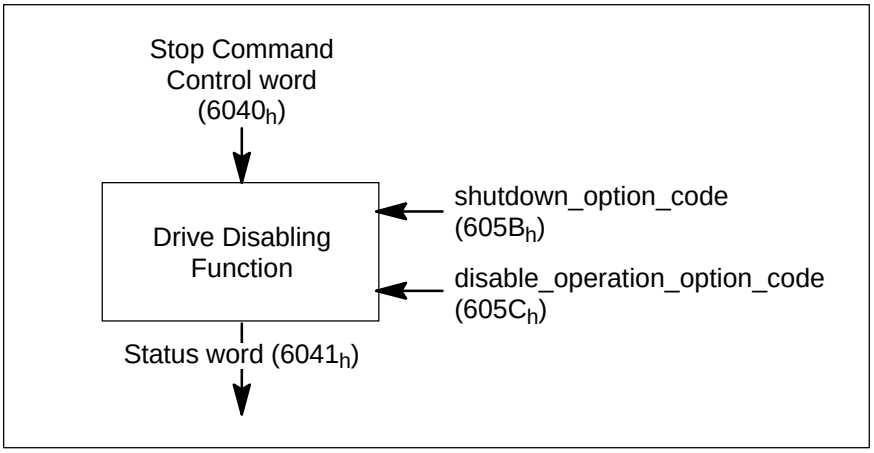
’Homing Mode’ is a special program function allowing the user to run complex of time-critical sequences, e. g. tool change or special reference operations, directly in the device.

The following objects are defined for management of the modes of operation:

- modes_of_operation
- modes_of_operation_display

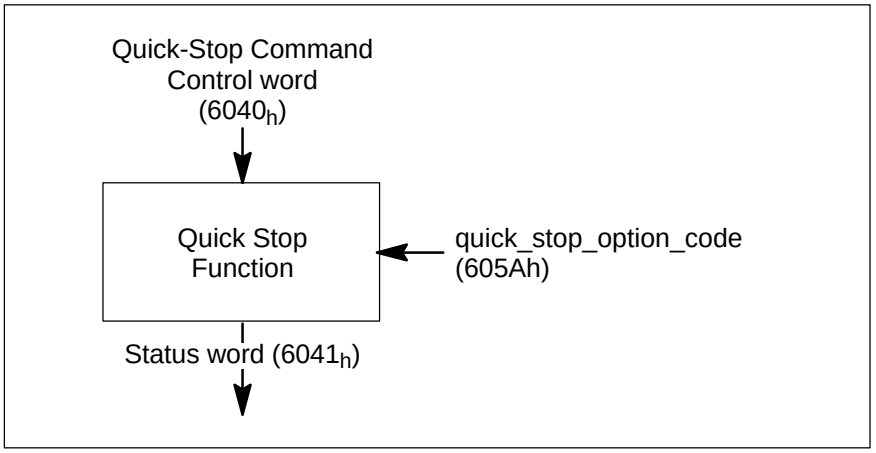
6.4.2 Disable Drive

This function defines the behaviour of the drive when transitioning from the ’Operation Enabled’ state to the ’Ready to Switch On’ (Shutdown command) or ’Switch On’ (Disable Operation command) state.



6.4.3 Quick-Stop Function

The Quick-Stop Function is triggered by the Quick-Stop command.



6.5 Fault Reaction

Fatal Faults

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

Non-Fatal Faults

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

If a fault occurs the drive will always enter the ’Fault’ state, even if the fault clears before the drive enters the ’Fault’ state. The ’Fault’ state can only be left with the command ’Reset Fault’ if no further fault exists on the drive.

Notes:

7 Factor Group

7.1 Meaning of the factors

The factors defined in the Factor Group are used to normalise physical values.

The factors are composed of:

- '**dimension index**'
  (defines the physical dimension of the object) and
- '**notation index**'
  The notation index is used in two ways:
  - For a unit **with decimal scaling** and notation index < 64
    the notation index defines the exponent/decimal place of the unit.
  - For a unit with **non-decimal scaling** and notation index > 64
    the notation index defines the sub-index of the physical dimension of the unit.

Examples for notation index < 64:

The value of the notation index is used as an exponent. The unit is defined by the physical dimension and calculated by unit type and exponent (see Table page 7–1).

position_units	dimension index = 1: length notation index = –6: micrometer position_units = $10^{notation\_index}$ x f(dimension_index) = 10^{-6} m
	dimension index = 12: angle notation index = 0: radian position_units = $10^{notation\_index}$ x f(dimension_index) = radian
velocity_units	dimension index = 13: speed notation index = –3: millimeter per second velocity_units = $10^{notation\_index}$ x f(dimension_index) = 10^{-3} m/s
frequency_units	dimension index = 28: frequency notation index = 3: kilohertz frequency_units = $10^{notation\_index}$ x f(dimension_index) = 10^{3} Hz

Example for notation index > 64:

The unit is defined by the physical dimension and the unit type (see Table page 7–1).

position_units	dimension index = 12: angle notation index = 76: minute position_units = f(dimension_index,notation_index) = minute

7.2 Object Description

Objects in this group represent factors which are necessary to normalise the physical inputs and outputs.

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
6089_h	VAR	position_notation_index	Integer8	rw
$608A_h$	VAR	position_dimension_index	Unsigned8	rw
$608B_h$	VAR	velocity_notation_index	Integer8	rw
$608C_h$	VAR	velocity_dimension_index	Unsigned8	rw
$608D_h$	VAR	acceleration_notation_index	Integer8	rw
$608E_h$	VAR	acceleration_dimension_index	Unsigned8	rw
6091_h	ARRAY	gear_ratio	Unsigned32	rw
6092_h	ARRAY	feed_constant	Unsigned32	rw
2567_h	ARRAY	position_encoder_resolution	–	rw

C-6089h position_notation_index

Is used to scale the following objects:

- position_actual_value
- position_demand_value
- target_position
- position_window
- following_error_window
- home_offset
- software_position_limit

Index	C-6089$_h$
Object code	VAR
Data type	Integer8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	–128..127 (for admissible values see Table page 7–3)	
Default value	77	
Substitute value	C-2580$_h$	

C-608Ah position_dimension_index

Is used to scale the following objects:

- position_actual_value
- position_demand_value
- target_position
- position_window
- following_error_window
- home_offset
- software_position_limit

Index	**C-608A$_h$**
Object code	VAR
Data type	Unsigned8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	1, 12 (see Table)	
Default value	12	
Substitute value	C-2580$_h$	

Position	**Unit 1 LSB =**	**position _dimension_index**	**position _notation_index**
Rotational	rad	12	0
	sec		75
	min		76
	degree		77
	gon		78
	10^{-3} degree		127
Translational	km	1	3
	m		0
	mm		–3
	µm		–6

C-608Bh velocity_notation_index

Is used to scale the following objects:

- velocity_actual_value
- velocity_demand_value
- profile_velocity
- velocity_window
- max_profile_velocity
- velocity_threshold
- target_velocity
- homing_speeds

Index	**C-608B$_h$**
Object code	VAR
Data type	Integer8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	–128..127 (for admissible values see Table page 7–5)	
Default value	73	
Substitute value	C-2581$_h$	

C-608Ch velocity_dimension_index

Is used to scale the following objects:

- velocity_actual_value
- velocity_demand_value
- profile_velocity
- velocity_window
- max_profile_velocity
- velocity_threshold
- target_velocity
- homing_speeds

Index	**C-608C$_h$**
Object code	VAR
Data type	Unsigned8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	11, 13 (see Table)	
Default value	11	
Substitute value	C-2581$_h$	

Speed	**Unit 1 LSB =**	**velocity _dimension_index**	**velocity _notation_index**
Rotational	s^{-1}	11	0
	min^{-1}		73
	h^{-1}		74
Translational	m/s	13	0
	mm/s		–3
	mm/min		79
	m/min		80
	km/min		81
	mm/h		82
	m/h		83
	km/h		84

C-608Dh acceleration_notation_index

Is used to scale the following objects:

- profile_acceleration
- profile_deceleration
- quick_stop_deceleration
- max_acceleration
- homing_acceleration

Index	**C-608D$_h$**
Object code	VAR
Data type	Integer8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	–128..127 (for admissible values see Table page 7–7)	
Default value	127	
Substitute value	C-2582$_h$	

C-608Eh acceleration_dimension_index

Is used to scale the following objects:

- profile_acceleration
- profile_deceleration
- quick_stop_deceleration
- max_acceleration
- homing_acceleration

Index	**C-608E$_h$**
Object code	VAR
Data type	Unsigned8

Value description

Object class	M: –	O: pc, pp, ip, hm, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	35 (see Table)	
Default value	35	
Substitute value	C-2582$_h$	

Acceleration	**Unit 1 LSB =**	**acceleration _dimension_index**	**acceleration _notation_index**
Rotational	rev/s^2	35	0
	10^{-3} rad/s^2		127
Translational	10^{-6} m/s^2	35	126

C-6091h gear_ratio

This object defines the gear ratio:

$$\text{gear\_ratio} = \frac{\text{motor\_revolutions}}{\text{shaft\_revolutions}}$$

Index	**C-6091$_h$**
Object code	ARRAY
Number of elements	2
Data type	Unsigned32

motor_revolutions

Value description

Subindex	**01$_h$**	
Object class	M: –	O: pc, p,p ip, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0..(2^{32} – 1)	
Default value	1	

shaft_revolutions

Value description

Subindex	**02$_h$**	
Object class	M: –	O: pc, pp, ip, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0..(2^{32} – 1)	
Default value	1	

C-6092h feed_constant

This object defines the feed constant as feed per shaft revolution. This includes the gear, if present.

$$\text{feed\_constant} = \frac{\text{feed [mm/rev]}}{\text{shaft\_revolutions}}$$

Index	**C-6092$_h$**
Object code	ARRAY
Number of elements	2
Data type	Unsigned32

feed

Value description

Subindex	**01$_h$**	
Object class	M: –	O: pc, pp, ip, pv, tq
Access	rw	
PDO mapping	no	
Units	mm/rev	
Value range	0..(2^{32} – 1)	
Default value	1	

shaft_revolutions

Value description

Subindex	**02$_h$**	
Object class	M: –	O: pc, pp, ip, pv, tq
Access	ro	
PDO mapping	no	
Units	–	
Value range	0..(2^{32} – 1)	
Default value	10.000	

C-2567h position_encoder_resolution

The drive calculates the encoder resolution in pulses per revolution using the number of motor encoder increments (encoder_increments) and the factor multiplication_1.

position_encoder_resolution = encoder_increments • multiplication_1

Index	**C-2567$_h$**
Object code	ARRAY
Number of elements	2

encoder_increments

Value description

Subindex	**01$_h$**	
Object class	M: –	O: pc, pp, ip, pv, tq
Access	ro	
PDO mapping	no	
Units	–	
Value range	0..(2^{32} – 1)	
Default value	0	
Data type	Integer32	

multiplication_1

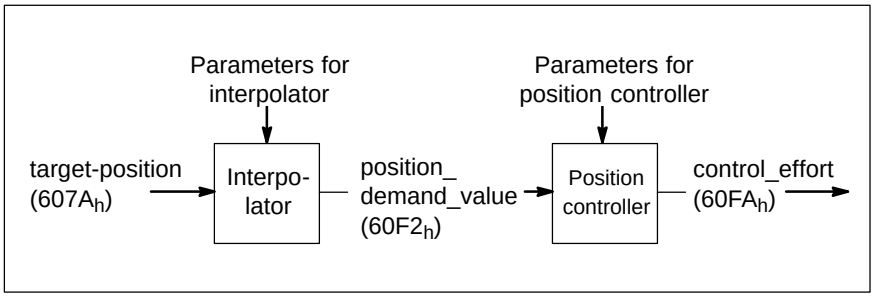
Value description

Subindex	**02$_h$**	
Object class	M: –	O: pc, pp, ip, pv, tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	4...512	
Default value	4	
Data type	Unsigned32	

8 Profile Position Mode

8.1 Overview

The overall structure of this mode is shown in the following.
From the object target_position, the interpolator generates the position_demand_value for the control loop described in ”Position Control Function” (Chapter 10). These two function blocks are optionally controlled by additional parameter sets.



At the input to the interpolator, objects may have optional limits applied before being normalised to internal units. Normalised objects are denoted with an asterisk.

Input Data Description

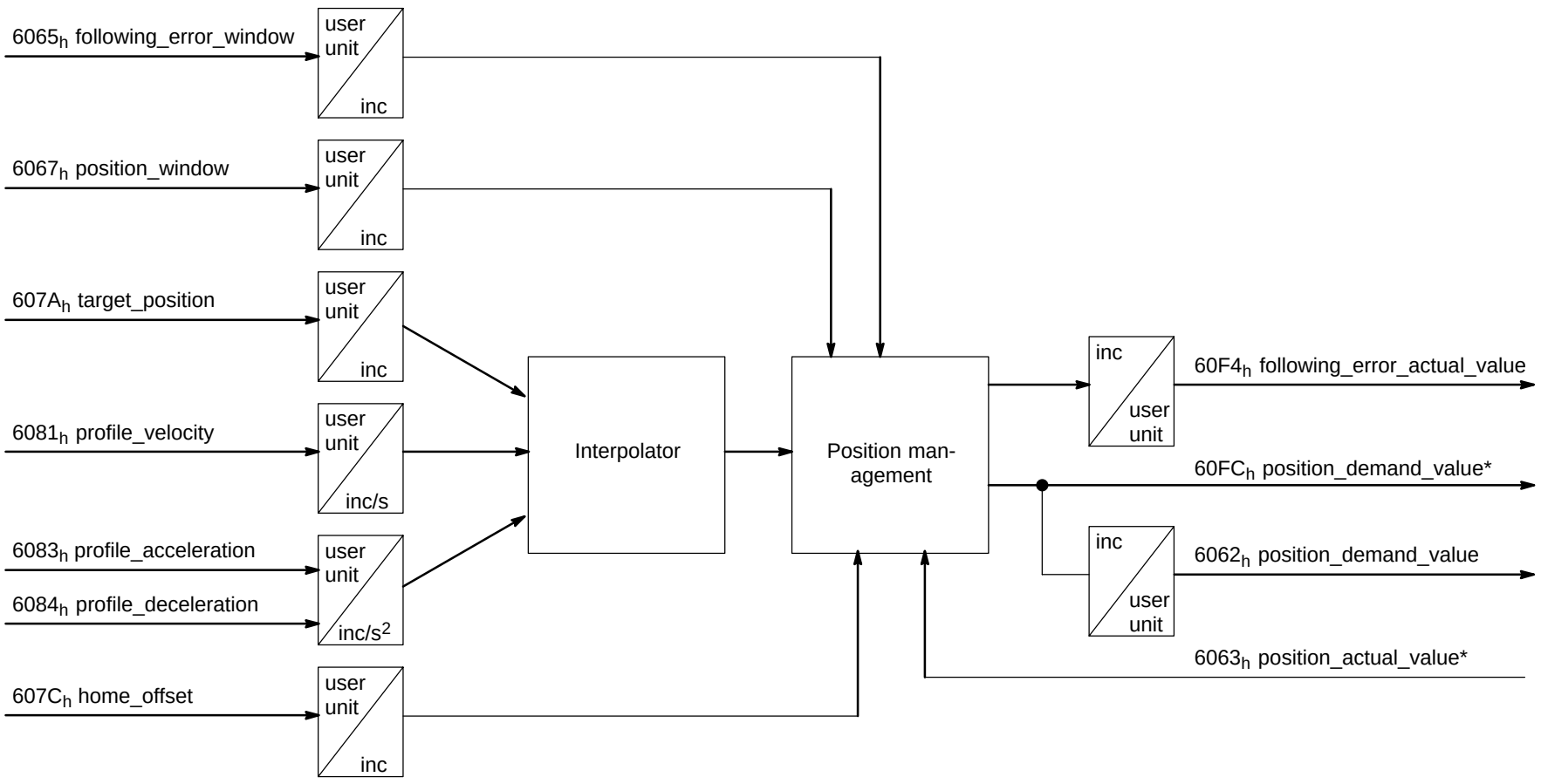
Mode of operation	Input parameters
Profile Position Mode	target_position, profile_velocity, profile_acceleration, profile_deceleration, quick_stop_deceleration, quick_stop_option_code, motion_profile_type, max_profile_velocity, max_motor_speed, software_position_limit

Output Data Description

The output value provided by the interpolator is the input for the position controller.

Mode of operation	Output parameters
Profile Position Mode	position_demand_value*

Profile Position Mode



* Values normalised

8.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
$607A_h$	VAR	target_position	Integer32	rw
$607D_h$	ARRAY	software_position_limit	Integer32	rw
$607F_h$	VAR	max_profile_velocity	Unsigned32	rw
6080_h	VAR	max_motor_speed	Unsigned16	rw
6081_h	VAR	profile_velocity	Unsigned32	rw
6083_h	VAR	profile_acceleration	Unsigned32	rw
6084_h	VAR	profile_deceleration	Unsigned32	rw
6085_h	VAR	quick_stop_deceleration	Unsigned32	rw
6086_h	VAR	motion_profile_type	Integer16	rw
$60C5_h$	VAR	max_acceleration	Unsigned32	rw

C-607A$_h$ target_position

This object contains the position value the drive moves to in the Profile Position Mode using the current settings of the objects velocity, acceleration, deceleration, motion_profile_type etc. The target_position is given in user defined position units and converted internally to position increments. The target_position is interpreted as absolute or relative depending on the position of bit 6 in the control_word.

Index	**C-607A$_h$**
Object code	VAR
Data type	Integer32

Value description

Object class	M: pp pc ip	O: –
Access	rw	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})..(2^{31}-1)$	
Default value	0	

C-607D$_h$ software_position_limit

This object defines the absolute position limits for:

- position_demand_value and
- position_actual_value.

Every new target_position is checked against these limits.
The values are specified in position units (same as target_position) and are always relative to the machine home position.

Index	**C-607D$_h$**
Object code	ARRAY
Number of elements	2
Data type	Integer32

min_position_limit

Sub-index	**01$_h$**	
Object class	M: –	O: pp, pc, ip
Access	rw	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

max_position_limit

Sub-index	**02$_h$**	
Object class	M: –	O: pp, pc, ip
Access	rw	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

C-607F$_h$ max_profile_velocity

Maximum permissible velocity for both directions. Input using the same units as profile_velocity.

Index	**C-607F$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: pp, ip, pv, tq
Access	rw	
PDO mapping	yes	
Units	Speed units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	Weighting-dependent	
Default value	Weighting-dependent	

C-6080$_h$ max_motor_speed

The maximum allowable motor speed in either direction is given in rpm. This value is used to protect the motor and is automatically taken from the electronic nameplate.

Index	**C-6080$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: all
Access	ro	
PDO mapping	no	
Units	rpm	
Value range	0..65535	
Default value	–	

C-6081$_h$ profile_velocity

This object describes the velocity attained at the end of the acceleration ramp during a profiled move. It is valid for both directions of motion and is given in speed units.

Index	**C-6081$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: pp, pv	O: –
Access	rw	
PDO mapping	yes	
Units	Speed units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	0...(2^{32}–1), motor- and weighting-dependent	
Default value	motor- and weighting-dependent	

C-6083$_h$ profile_acceleration

This object defines the acceleration ramp. Input in user defined acceleration units.

Index	**C-6083$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: pp, pv	O: –
Access	rw	
PDO mapping	possible	
Units	Acceleration units, as in acceleration_notation_index (C-608D) and acceleration_dimension_index (C-608E)	
Value range	0...(2^{32}–1)	
Default value	Weighting-dependent	

C-6084$_h$ profile_deceleration

This object describes the deceleration ramp. Input as for profile_acceleration.

Index	**C-6084$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: pp, pv	O: –
Access	rw	
PDO mapping	yes	
Units	Acceleration units	
Value range	0...(2^{32} –1)	
Default value	Weighting-dependent	

C-6085$_h$ quick_stop_deceleration

This object describes the deceleration ramp for quick stop. Input as for profile_acceleration.

Index	**C-6085$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: pp, ip, pv, vo, tq, hc
Access	rw	
PDO mapping	yes	
Units	Acceleration units	
Value range	0...(2^{32} –1)	
Default value	Weighting-dependent	

C-6086$_h$ motion_profile_type

This object selects a linear ramp as type of motion.

Index	C-6086$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: pp, pv	O: –
Access	ro	
PDO mapping	no	
Units	none	
Value range	0 = linear ramp, trapezoidal profile	
Default value	0	

C-60C5$_h$ max_acceleration

This object limits the acceleration and deceleration behaviour of the motor to a value preventing the motor and machine from being destroyed. Max_acceleration is given in user defined units.

Index	C-60C5$_h$
Object code	VAR
Data type	Unsigned32

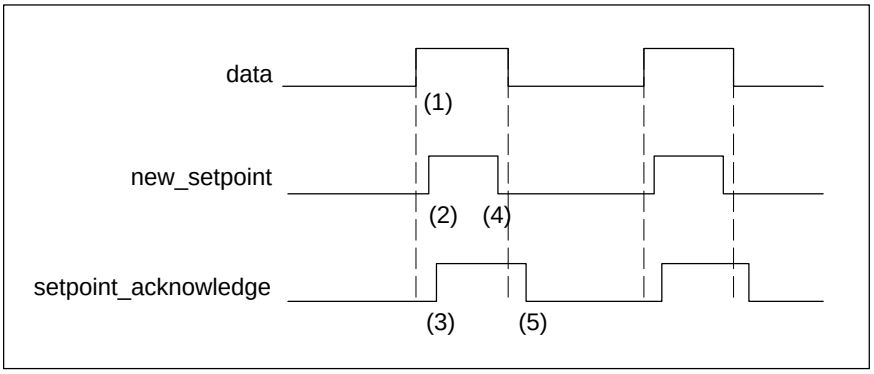
Value description

Object class	M: –	O: ip, pp, pc
Access	rw	
PDO mapping	yes	
Units	Acceleration units, as in acceleration_notation_index (C-608D) and acceleration_dimension_index (C-608E)	
Value range	0...(2^{32} –1)	
Default value	Weighting-dependent	

8.3 Functional Description

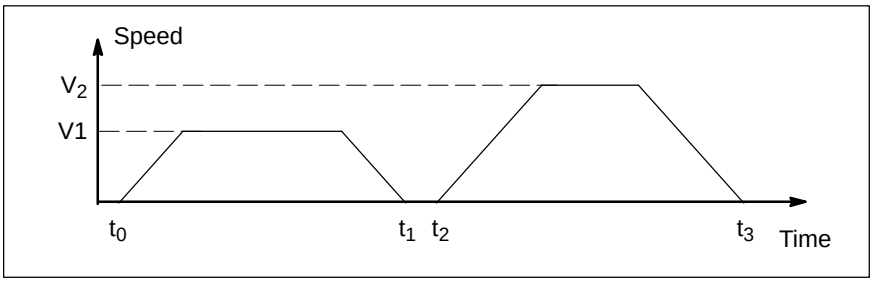
In the Profile Position Mode, target_positions are specified to the drive as individual setpoint values.
After reaching the target_position the drive signals this status to a host controller and then receives a new setpoint. After reaching the target_position the velocity is reduced to zero before starting a move to the next setpoint.

This mode of operation is controlled by the bits new_setpoint in the control_word and setpoint_acknowledge in the status_word. Thus a request – response mechanism can be set up in order to process the setpoints one after the other.



After data is applied to the drive (1), the controller signals that the data is valid by changing the bit new_setpoint to ”1” in the control_word (2). The drive responds with setpoint_acknowledge set to ”1” in the status_word (3) after it recognised the new valid data.
Now the controller may release new_setpoint (4) and afterwards the drive signals with setpoint_acknowledge equal ”0” its ability to accept new data again (5).

In the following figure this mechanism results in a velocity of zero after ramping down in order to reach a target_position x1 at t1. After signalling to the controller, that the setpoint is reached as described above, the next target_position x2 is processed at t2 and reached at t3.



9 Homing Mode

9.1 Overview

The Homing Mode offers different methods for the drive to travel to home position. A detailed description of the methods available is given in Section 9.3.

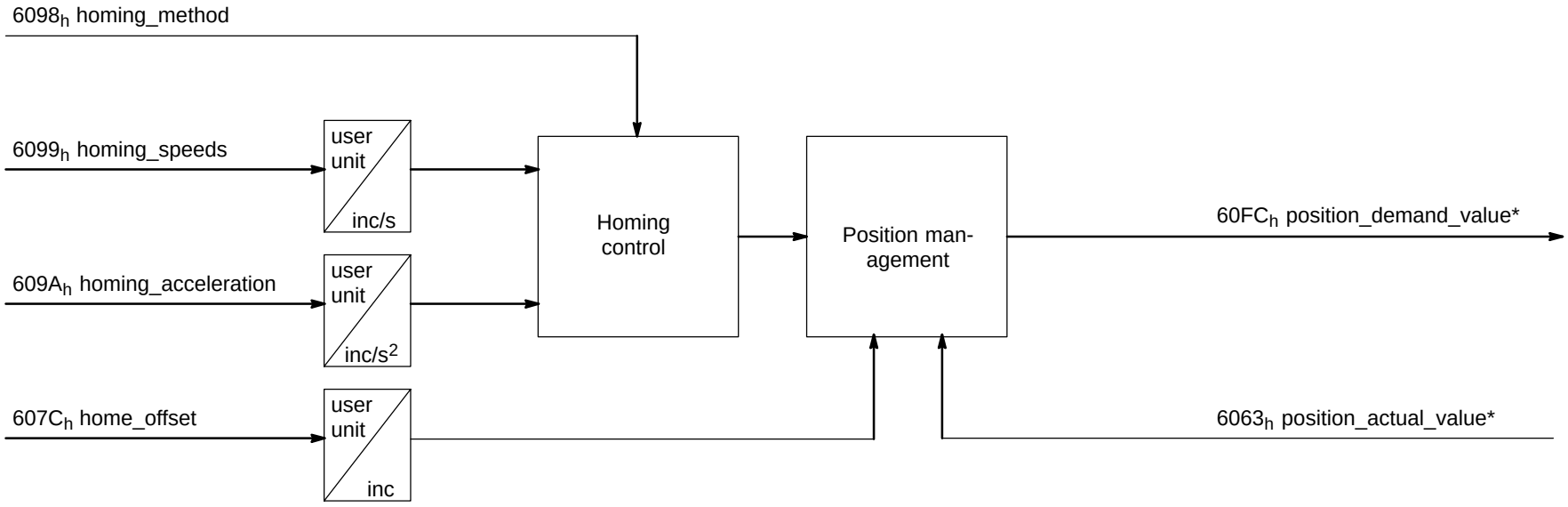
Input Data Description

Homing is specified via the objects homing_speeds, homing_acceleration and homing_method. Using the object home_offset, it is also possible to displace the zero point from the home position.

Output Data Description

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

Homing Mode



* Values normalised

9.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
2190	VAR	home_switch	Unsigned16	ro
$607C_h$	VAR	home_offset	Integer32	rw
6098_h	VAR	homing_method	Integer8	rw
6099_h	ARRAY	homing_speeds	Unsigned32	rw
$609A_h$	VAR	homing_acceleration	Unsigned32	rw

C-2190$_h$ homing_switch

The signal of a home switch connected at the digital input is evaluated via this object. For this purpose, enter this object in C-47D0$_h$ (inport_configuration_list).

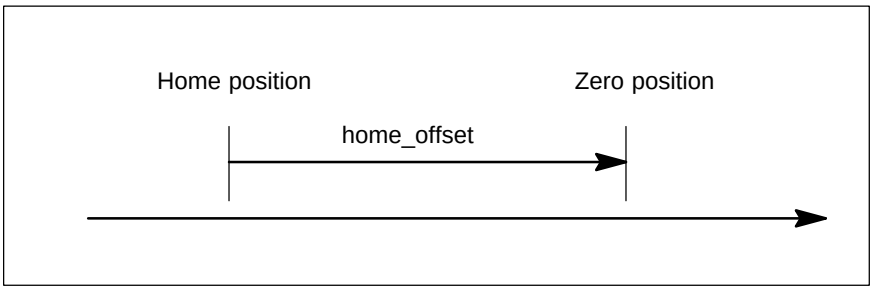
Index	C-2190$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: hm	O: –
Access	ro	
PDO mapping	yes	
Units	–	
Value range	0 = home switch activated 1= home switch not activated	
Default value	0	

C-607C$_h$ home_offset

This object describes the difference between the zero position of the application and the machine home position. 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 are relative to this new zero position.



Index	C-607C$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: hm
Access	rw	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

C-6098$_h$ homing_method

This object specifies the method that will be used during homing.

Index	C-6098$_h$
Object code	VAR
Data type	Integer8

Value description

Object class	M: hm	O: –
Access	rw	
PDO mapping	yes	
Units	–	
Value range	0 = no homing operation required 1...35 = methods 1 to 35 (for available methods, see page 9–7 ff.)	
Default value	0	

C-6099$_h$ homing_speeds

In this object the homing speed is specified.
No difference is made between 'speed_during_search_for_switch' and 'speed_during_search_for_zero', the same value is used for both objects.

Index	**C-6099$_h$**
Object code	ARRAY
Number of elements	2
Data type	Unsigned32

speed_during_search_for_switch

Sub-index	**01$_h$**	
Object class	M: hm	O: –
Access	rw	
PDO mapping	yes	
Units	Speed units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	0...(2^{32} – 1)	
Default value	0	

speed_during_search_for_zero (= C-6099.01)

Sub-index	**02$_h$**	
Object class	M: hm	O: –
Access	rw	
PDO mapping	no	
Units	see above	
Value range	0...(2^{32} – 1)	
Default value	0	

C-609A$_h$ homing_acceleration

This object establishes the acceleration and deceleration values effective during homing.

Index	C-609A$_h$
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: hm
Access	rw	
PDO mapping	yes	
Units	Acceleration units, as in acceleration_notation_index (C-608D) and acceleration_dimension_index (C-608E)	
Value range	0...(2^{31} –1)	
Default value	0	

9.3 Functional Description

For the operation of positioning, the drive has to know the axle position. If no absolute encoder is used, a homing operation of the axle is necessary. This takes place in the Homing Mode.

In the objects of the Homing Mode, the conditions and methods of homing are established. The object homing_method establishes the following:

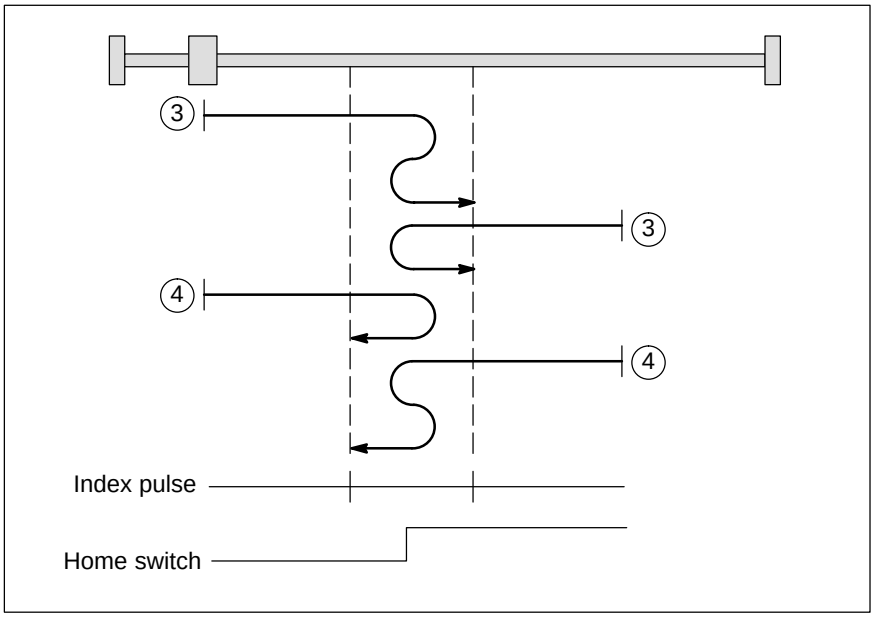
- Homing always using the home switch
- Direction of actuation
- with / without index pulse

In the following diagrams of the available homing methods, the different procedures are illustrated. The encircled number indicated the code of the homing_method.

9.3.1 Homing Methods

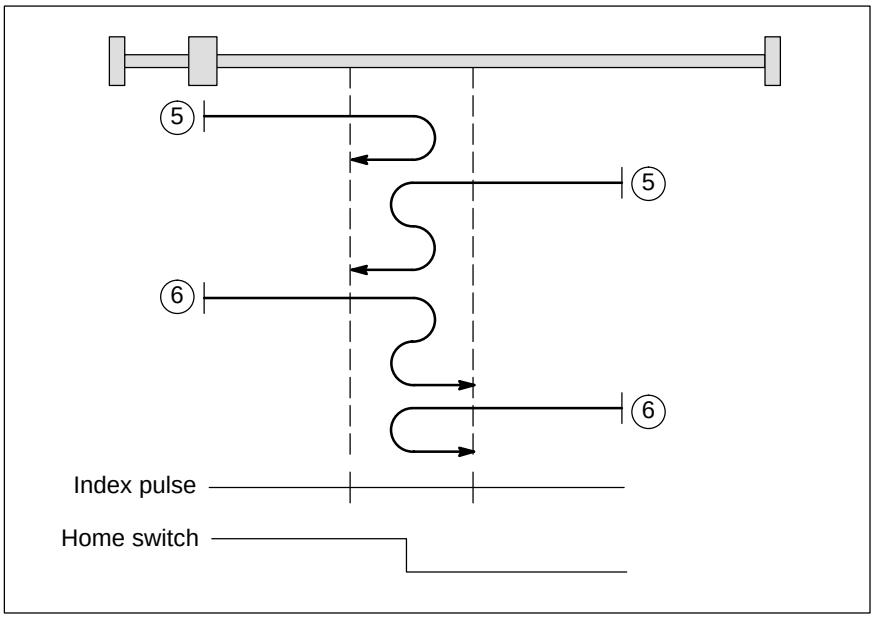
Methods 3, 4: Homing on the index pulse with positive home switch edge

The initial direction of movement is dependent on the state of the home switch. The home position is the first index pulse either to the left or to the right of the home switch.
If the initial axle position is so sited that the direction of movement must reverse during homing, the reversal takes place after the change of state of the home switch.



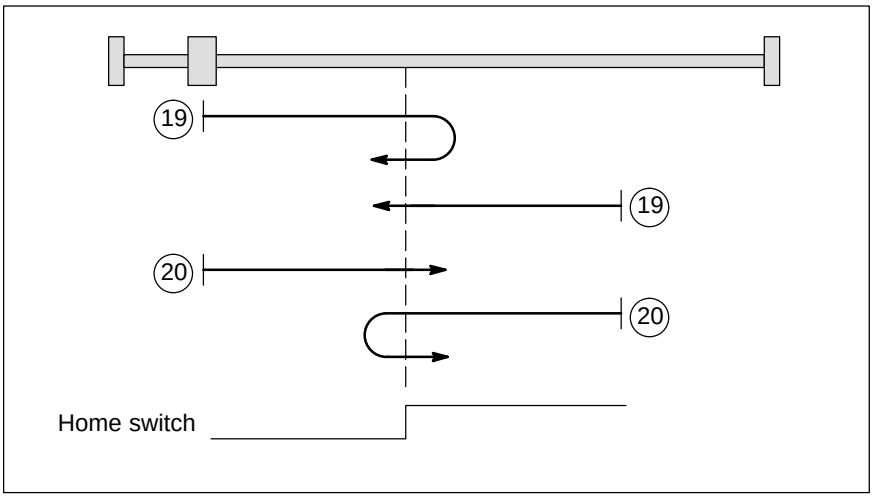
Methods 5, 6: Homing on the index pulse with negative home switch edge

The initial direction of movement is dependent on the state of the home switch. The home position is the first index pulse either to the left or to the right of the home switch.
If the initial axle position is so sited that the direction of movement must reverse during homing, the reversal takes place after the change of state of the home switch.



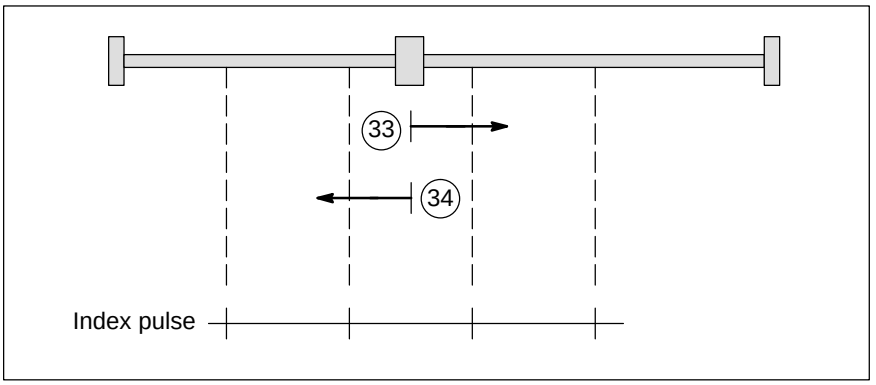
Methods 19 to 22: Homing without index pulse

These methods are similar to methods 3, 4 and 5, 6 except that the home position is not dependent on the index pulse but only dependent on the relevant home switch.



Methods 33, 34: Homing on the index pulse without home switch

The direction of homing is either negative or positive. The home position is the next index pulse in travel direction.



9.3.2 Sequence control via control_word and status_word

The Homing Mode is started by setting bit 4 in the control_word. The successful completion is indicated by bit 12 in the status_word. A one in bit 13 of the status_word indicates an error during the homing operation. The cause is found by reading the error codes.

control_word

Bit 4	Meaning
0	Homing operation inactive
0 → 1	Start homing operation
1	Homing operation active
1 → 0	Interrupt homing operation

status_word

Bit 13	Bit 12	Meaning
0	0	Homing operation not yet completed
0	1	Homing operation carried out successfully

Notes:

10 Position Control Function

10.1 Overview

The objects ’position_demand_value’ as an output value of the interpolator and ’position_actual_value’ from the motor encoder are the input values for the position control.
The behaviour of the control may be influenced by external control parameters.

All values are transformed, if necessary, from user defined units to normalised units through the objects of the Factor Group.

Input Data Description

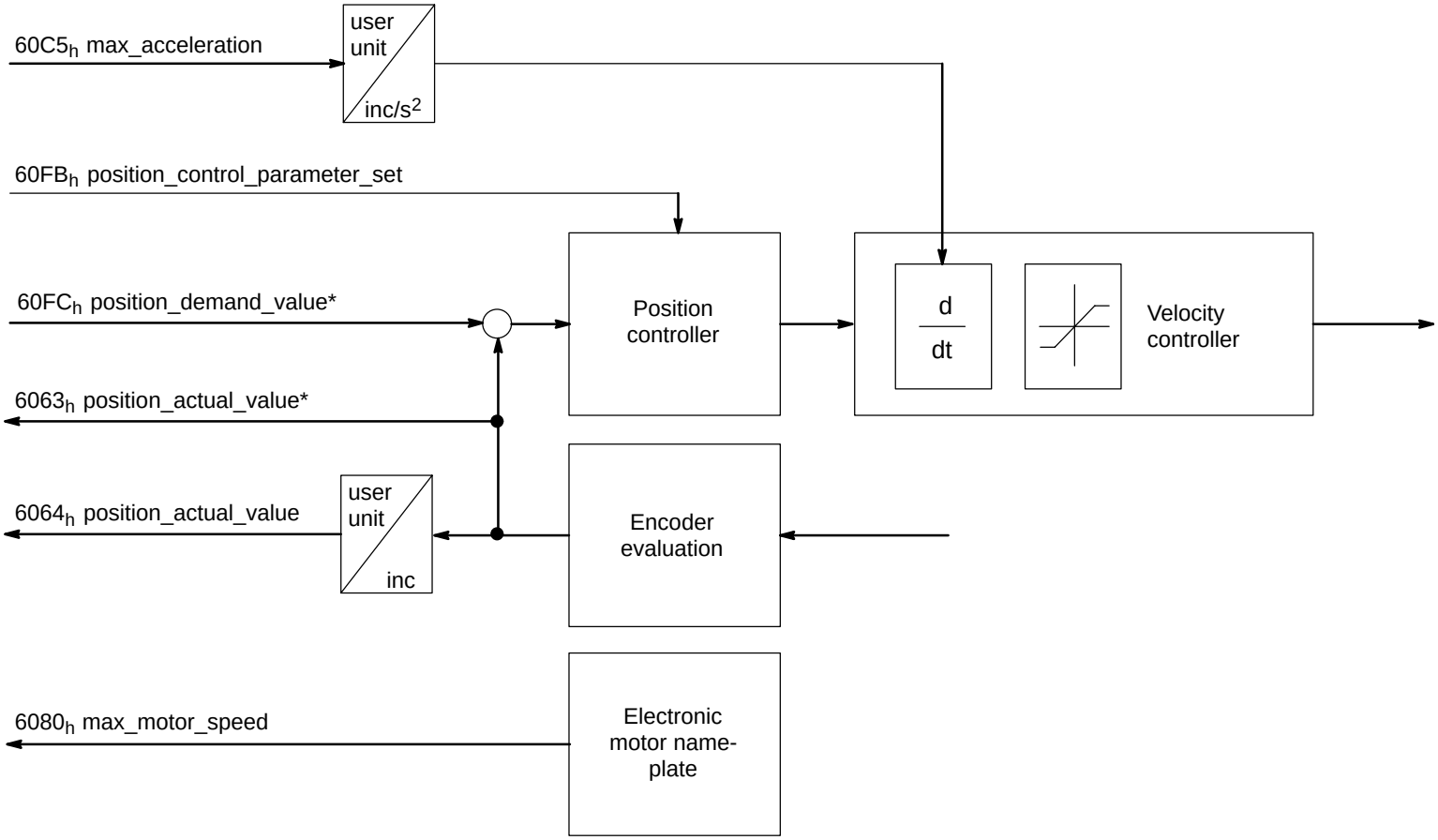
Depending on the active operation mode, only some of the mentioned input parameters may be necessary.

Mode of Operation	Input parameters
Position Mode	position_demand_value, position_window, following_error_window, position_actual_value, digital_inputs, target_position, position_range_limit
Homing Mode	
Interpolated Position Mode	

Output Data Description

Mode of Operation	Output parameters
Position Mode	status_word, digital_outputs
Homing Mode	
Interpolated Position Mode	

Profile Position Mode + Position Control Function



* Values normalised

10.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
$607A_h$	VAR	target_position	Integer32	8
$607C_h$	VAR	home_offset	Integer32	9
$607D_h$	VAR	software_position_limit	Integer32	8
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
2067_h	RECORD	modulo_control_parameters	–	rw
$209F_h$	VAR	monitoring_window	Unsigned16	rw
$20AF_h$	VAR	displacement_parameter	Integer32	rw
$20B1_h$	VAR	absolute_dimension_offset	Integer32	rw
4007_h	VAR	position_generator_cycle_time	Unsigned16	rw
$401F_h$	VAR	absolute_rotation_offset_1	Integer32	rw
4020_h	VAR	rotation_offset_command	Unsigned16	rw
$41F4_h$	VAR	velocity_feed_forward_coefficient	Unsigned16	rw
6062_h	VAR	position_demand_value	Integer32	ro
6063_h	VAR	position_actual_value*	Integer32	ro
6064_h	VAR	position_actual_value	Integer32	ro
6065_h	VAR	following_error_window	Unsigned32	rw
6067_h	VAR	position_window	Unsigned32	rw
$60F4_h$	VAR	following_error_actual_value	Integer32	ro
$60FB_h$	RECORD	position_control_parameter_set	–	rw
$60FC_h$	VAR	position_demand_value*	Integer32	rw

C-2067$_h$ modulo_control_parameters

Sub-index 01 modulo_value determines the position from which the drive must perform a modulo calculation. Thus, the current position is permanently transformed to a position range between 0 and the value entered (e.g. for endless axles).
For this purpose, the modulo calculation must be activate with sub-index 02.

Index	**C-2067$_h$**
Object code	RECORD
Number of Elements	2

modulo_value

Sub-index	**01$_h$**	
Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	
Data type	Integer32	

modulo_activate

Sub-index	**02$_h$**	
Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = no modulo calculation 1 = modulo calculation activated	
Default value	0	
Data type	Unsigned16	

C-206C$_h$ feedrate_override

The feedrate overdrive has a multiplying effect (unit of measure: %) on the speed setpoints calculated by the drive.

Index	**C-206C$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	yes	
Units	%	
Value range	0...120	
Default value	100	

C-209F$_h$ monitoring_window

Monitoring of the speed setpoint by evaluating the deviation between actual value and position setpoint. When the monitoring window (entry in % of maximum speed) is exceeded, the drive sets the error message 'excessive controller deviation'.

Index	**C-209F$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	0.1 %	
Value range	0...5000	
Default value	1200	

C-20AF$_h$ displacement_parameter

In this object, the drive saves the difference between the old position system and the referenced position system.

Index	C-20AF$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

C-20B1$_h$ absolute_dimension_offset

Distance from machine zero position to zero point of motor encoder with absolute measurement.

Index	C-20B1$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

C-4007$_h$ position_generator_cycle_time

Cycle time of the position setpoint generator.

Index	C-4007$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	ms	
Value range	2...8	
Default value	4	

C-401F$_h$ absolute_rotation_offset_1

Distance between machine zero position in terms of an absolute number of revolutions, in case of axles with a multi-turn absolute encoder.

Index	C-401F$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	revolutions	
Value range	–4095...+4095	
Default value	0	

C-4020$_h$ rotation_offset_command

Command for determination of the absolute_rotation_offset (C-401F$_h$).

Index	**C-4020$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = no command 3 = execute command	
Default value	0	

C-41F4$_h$ velocity_feed_forward_coefficient

This object reduces the following distance between position setpoint and position actual value at constant speed.
100 % ≙ Following distance ”0”

Index	**C-41F4$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	0.1 %	
Value range	0...1100	
Default value	800	

C-6062$_h$ position_demand_value

Index	C-6062$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc
Access	ro	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	–	

C-6063$_h$ position_actual_value*

The normalised actual value is one of the two input values of the closed loop position control. The unit is defined as increment and can be transformed from user defined units to increments with the position_factor, if necessary.

Index	C-6063$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc, pp, ip, hm, tq
Access	ro	
PDO mapping	yes	
Units	increments	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	–	

C-6064$_h$ position_actual_value

This object is the position actual value in user defined units.

Index	**C-6064$_h$**
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc, pp, ip, hm, tq
Access	ro	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	–	

C-6065$_h$ following_error_window

This object defines a range of tolerated positions symmetrically to the position_demand_value for the purpose of following error detection. If the object position_actual_value is outside of the range, a following error is detected.

A following error might occur when:

- a drive is blocked
- unreachable profile velocity
- wrong closed loop coefficients.

Index	**C-6065$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: pc, pp
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$0...(2^{32}-1)$	
Default value	0 = monitoring switched off	

C-6067$_h$ position_window

This object defines a range of tolerated positions symmetrically to the object target_position for the purpose of in-position detection. If the position actual value is within this range, target_position is regarded as reached.

Index	**C-6067$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: pc
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	0...(2^{32} –1)	
Default value	weighting-dependent	

C-60F4$_h$ following_error_actual_value

This object contains the actual value of the following error.

Index	**C-60F4$_h$**
Object code	VAR
Data type	Integer32

Value description

Object class	M: –	O: pc, pp, ip, hm, tq
Access	ro	
PDO mapping	yes	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	(–2^{31})...(2^{31} –1)	
Default value	–	

C-60FB$_h$ position_control_parameter_set

Sub-index 1 and 2 describe the KV factor of the position control.

$$KV = \frac{\text{position\_loop\_kv\_factor\_numerator [(m/min)/mm]}}{\text{position\_loop\_kv\_factor\_denominator}}$$

The KV factor defines the gain of the position control loop over the whole speed range. Increasing the KV factor decreases the following distance. Calculate the KV factor to be entered using the following equation:

$$KV = 1 / t_m * 16.67$$

t_m : Acceleration of the drive under load (with connected mechanical axle system and average load during processing) to max. speed in seconds.

Index	**C-60FB$_h$**
Object code	RECORD
Number of Elements	2
Data type	Unsigned16

position_loop_kv_factor_numerator

Sub-index	**01$_h$**	
Object class	M: –	O: pc, pp, ip, hm
Access	rw	
PDO mapping	no	
Units	(m/min)/mm	
Value range	0...655	
Default value	1	

position_loop_kv_factor_denominator

Sub-index	**02$_h$**	
Object class	M: –	O: pc, pp, ip, hm
Access	rw	
PDO mapping	no	
Units	–	
Value range	0...100	
Default value	1	

C-60FC$_h$ position_demand_value*

This normalised output value of the interpolator in Position Mode is an internal value. To reduce calculation time for some applications, this object can be used in addition to the position_demand_value (6062$_h$).

Index	**C-60FC$_h$**
Object code	VAR
Data type	Integer32

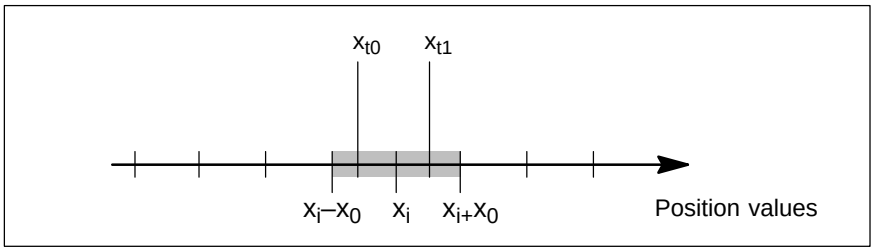
Value description

Object class	M: –	O: pc, pp, ip, hm, tq
Access	ro	
PDO mapping	yes	
Units	increments	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	–	

10.3 Functional Description

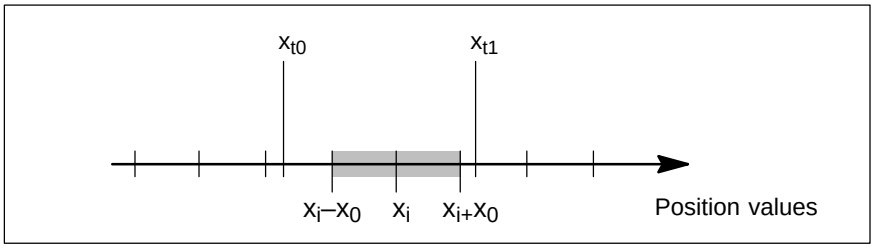
In position

Around the target_position x_i a symmetrical window is defined from x_i-x_0 to x_i+x_0. If the position actual values are situated within this range, as e.g. x_{t0} and x_{t1}, the drive sets bit 10 (target_reached) in the status_word.



Following error

The symmetrical window following_error_window may be defined to detect a following error. If the position actual value is situated outside of this range, as e.g. x_{t0} and x_{t1}, the drive sets bit 13 (following error) in the status_word.



Notes:

11 Interpolated Position Mode

11.1 Overview

The Interpolated Position Mode is used for the coordinated movement of interdependent axles or to move individual axles with time interpolation of setpoints. The Interpolated Position Mode normally uses time synchronisation mechanisms like the synchronisation object for a time coordination of the related drive inverters.

The object interpolation_data_record contains the interpolation data, the data record size is fixed in the object size_of_data_record, a sub-index of interpolation_data_configuration.

For synchronous operation the interpolation cycle time is defined by the object interpolation_time_period. Time synchronisation is done via the CANopen synchronisation telegram.

The interpolation algorithm is set at linear interpolation. For this purpose the drive takes one interpolation data setpoint and calculates a position_demand_value for each interpolation cycle by interpolating position over a certain period of time.
Optionally the common limit functions for speed, acceleration and deceleration may be applied to the interpolation data.

Input Data Description

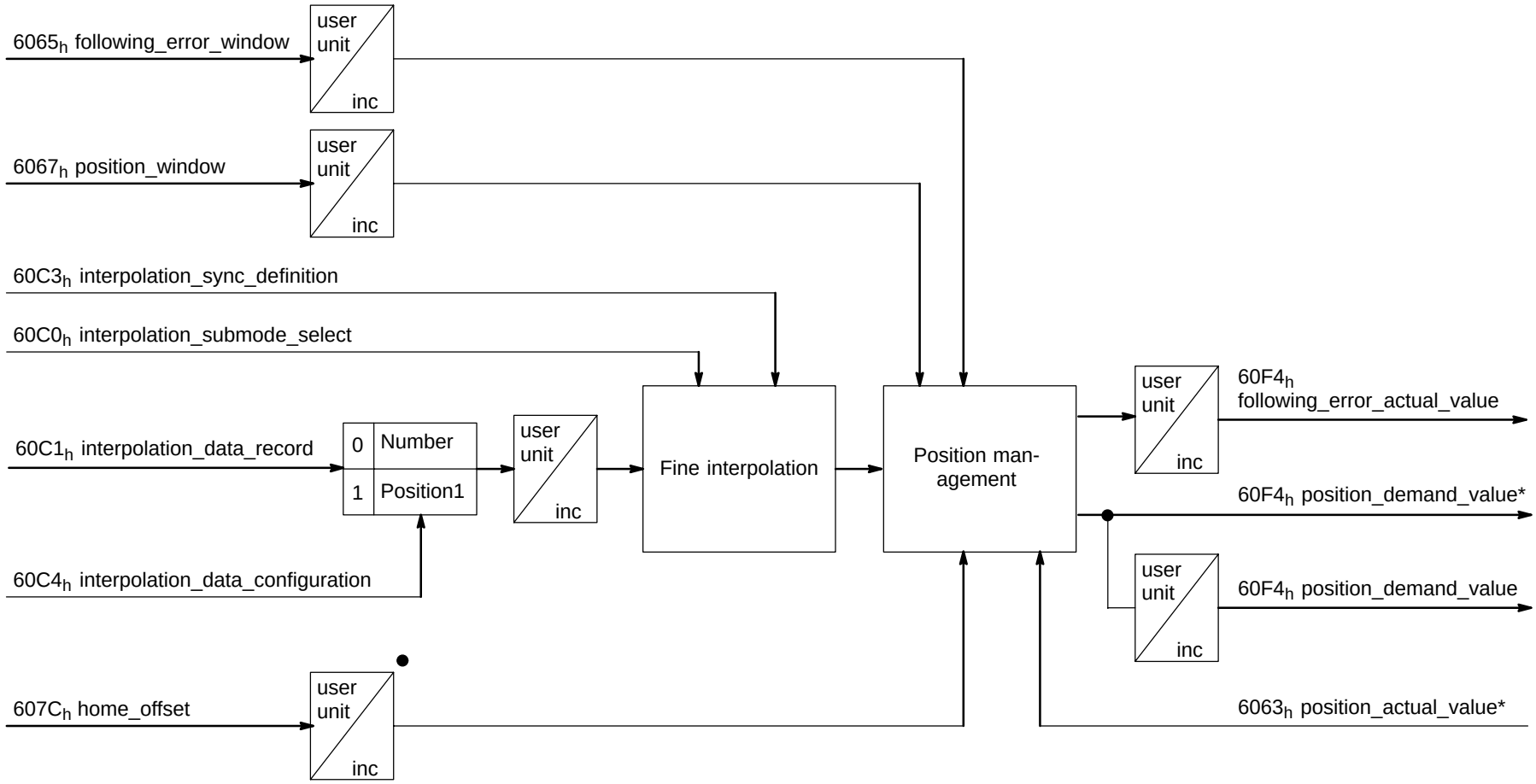
Mode of Operation	Input parameters
Interpolated Position Mode	interpolation_submode_select, max_profile_velocity, profile_acceleration, profile_deceleration, quick_stop_deceleration, quick_stop_mode

Output Data Description

Output for linear time interpolation.

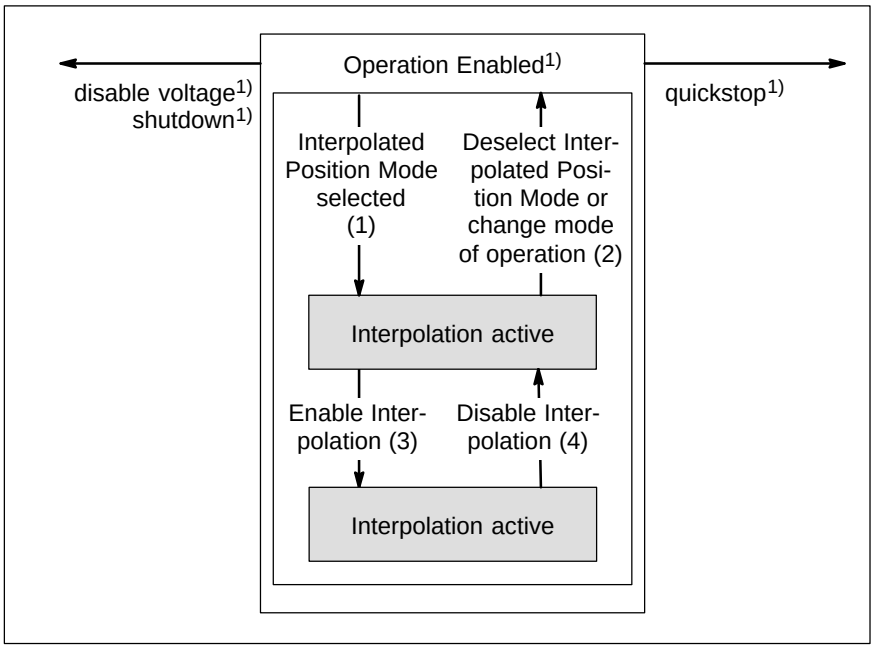
Mode of Operation	Output parameter
Interpolated Position Mode	position_demand_value*

Interpolated Position Mode



* Values normalised

11.1.1 Internal States



1) see State Machine

Interpolation inactive

This state is entered when the Interpolated Position Mode is selected in state Operation Enabled. The drive will accept input data and will buffer it for interpolation calculations, but it does not move the axles.

Interpolation active

This state is entered when the drive is in state Operation Enabled, the Interpolated Position Mode is selected and enabled. The drive unit will accept input data and it moves the axles.

State Transitions

(1) Interpolated Position Mode deselected→ Interpolated Position Mode selected

Event: Select Interpolated Position Mode with control_word in state Operation Enabled.

(2) Interpolated Position Mode inactive→ Interpolated Position Mode deselected

Event: Select any other mode of operation in control_word in state Operation Enabled.

(3) Interpolated Position Mode inactive→ Interpolated Position Mode active

Event: Set bit 4 enable_ip_mode in control_word in state Operation Enabled with Interpolated Position Mode selected.

(4) Interpolated Position Mode inactive→ Interpolated Position Mode active

Event: Reset bit 4 enable_ip_mode in control_word in state Operation Enabled with Interpolated Position Mode selected.

11.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6
6062_h	VAR	position_demand_value	Integer32	10
6063_h	VAR	position_actual_value*	Integer32	10
$606A_h$	VAR	sensor_selection_code	Unsigned8	12
$607F_h$	VAR	max_profile_velocity	Unsigned32	8
$60C5_h$	VAR	max_acceleration	Unsigned32	8

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
$60C0_h$	VAR	interpolation_submode_select	Integer16	rw
$60C1_h$	RECORD	interpolation_data_record	Integer32	rw
$60C3_h$	ARRAY	interpolation_sync_definition	Unsigned8	rw
$60C4_h$	RECORD	interpolation_data_configuration	Unsigned16	rw

C-60C0$_h$ interpolation_submode_select

This object specifies a linear interpolation.

Index	C-60C0$_h$
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: ip
Access	ro	
PDO mapping	no	
Units	–	
Value range	0 = linear interpolation	
Default value	0	

C-60C1$_h$ interpolation_data_record

This object contains the interpolation data record to be processed. It is regarded as new position setpoint.

Index	C-60C1$_h$
Object code	RECORD
Number of elements	1
Data type	Interpolated Position Mode dependent

X1: the first parameter of ip function $f_{ip}(x_1, .. x_n)$

Sub-index	01$_h$	
Object class	M: –	O: ip
Access	rw	
PDO mapping	possible	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	$(-2^{31})...(2^{31}-1)$	
Default value	0	

C-60C3$_h$ interpolation_sync_definition

Communication object for synchronisation with other devices.

Synchronisation can be effected by the general synchronisation signal as described in CiA/DS-301, or by a specific group synchronisation signal. The devices synchronise either after each reception of a trigger signal or after a specified number of occurrences of this trigger signal.

Index	**C-60C3$_h$**
Object code	ARRAY
Number of elements	2
Data type	Unsigned8

synchronize_on_group

Sub-index	**01$_h$**	
Object class	M: –	O: ip
Access	ro	
PDO mapping	no	
Units	number	
Value range	0 = use general synchronisation signal	
Default value	0	

ip_sync_every_n_event

Sub-index	**02$_h$**	
Object class	M: –	O: ip
Access	ro	
PDO mapping	no	
Units	counts	
Value range	1 = synchronise after each reception	
Default value	1	

C-60C4$_h$ interpolation_data_configuration

The size of the input data buffer as maximum number of data to be received.

Index	**C-60C4$_h$**
Object code	RECORD
Number of elements	1

max_buffer_size

Sub-index	**01$_h$**	
Object class	M: –	O: ip
Access	ro	
PDO mapping	no	
Units	number	
Value range	1	
Default value	1	
Data type	Unsigned32	

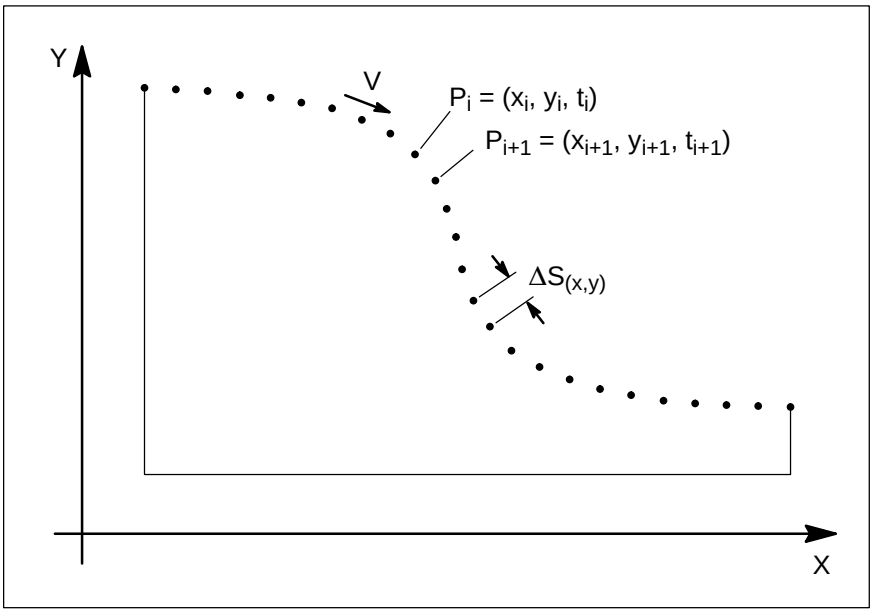
11.3 Functional Description

The drive is controlled and supervised by the control_word and the status_word. To choose the operation mode, modes_of_operation is used, and the operation mode is monitored by modes_of_operation_display.
New data are processed after the next synchronisation signal.

Linear Interpolation with several Axles

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

During operation with several axles the host calculates the next position and timestamp, and transmits them to the different axles. For each setpoint P_i the interpolator has to calculate x_i, y_i and t_i. Each drive receives an object interpolation_data_record which is processed independent from the other drives.



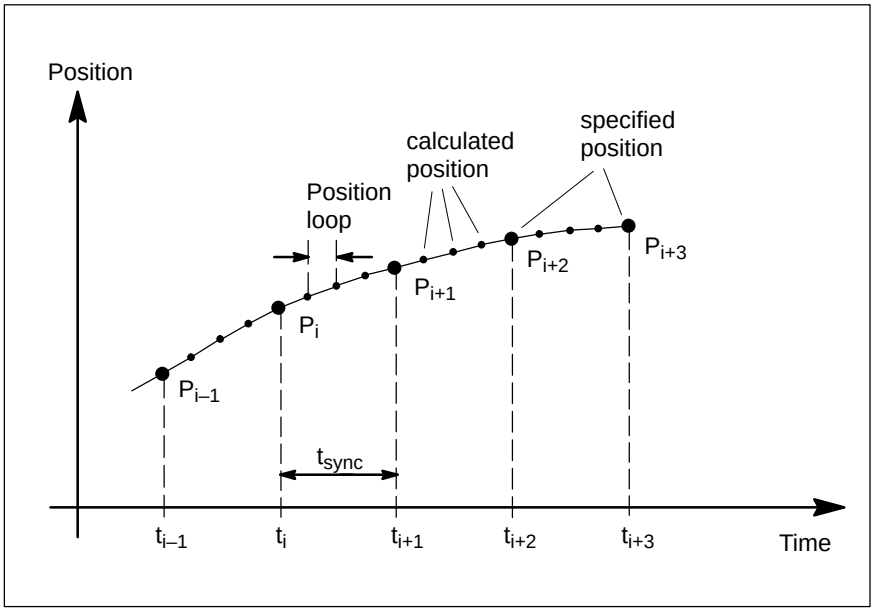
In a **centralised drive system** with the interpolation performed locally in the drive, a central clocking scheme for synchronisation of the different axles based on any kind of synchronisation signal is used. This results is a movement depending on the calculation cycle time of the interpolation controller. The velocity becomes more or less a fixed value for each axle.

Position calculation in Interpolated Position Mode for several axles:

Calculated Positions	**ip_data_records for**		
	axle x	**axle y**	**axle z**
P_i	x_i , t_i	y_i , t_i	z_i , t_i
P_{i+1}	x_{i+1} , t_{i+1}	y_{i+1} , t_{i+1}	z_{i+1} , t_{i+1}
P_{i+2}	x_{i+2} , t_{i+2}	y_{i+2} , t_{i+2}	z_{i+2} , t_{i+2}
P_{i+3}	x_{i+3} , t_{i+3}	y_{i+3} , t_{i+3}	z_{i+3} , t_{i+3}
•	•	•	•
•	•	•	•
•	•	•	•
P_{i+n}	x_{i+n} , t_{i+n}	y_{i+n} , t_{i+n}	z_{i+n} , t_{i+n}

In a **decentralised drive system**, a host starts all relevant axles by changing their internal state to interpolation active after preparing and sending the object interpolation_data_records to all axles and synchronising them by a (group) synchronisation signal.
Each axle calculates internally and independently the necessary speed and acceleration needed to move from one position to the next. Along this track every axle controls the movement between the setpoints independently from the other axles.

Using this information, every axle can perform linear interpolation:



Notes:

12 Profile Velocity Mode

12.1 Overview

The Profile Velocity Mode includes the following sub-functions:

- Setpoint specification
- Velocity capture using position sensor or velocity sensor
- Velocity control with appropriate input and output signals
- Limitation of torque_demand_value
- Monitoring of the target_velocity using a window function
- Monitoring of the velocity_actual_value using a threshold

Reference value generator and its input parameters, see chapter 8:

- target_velocity
- profile_acceleration
- emergency_stop and
- motion_profile_type

The speed is captured via a resolver or encoder in the motors.
The velocity controller calculates a torque variable which is limited to torque_max_value. The limited total is used as input to the torque controller as a torque_demand_value.

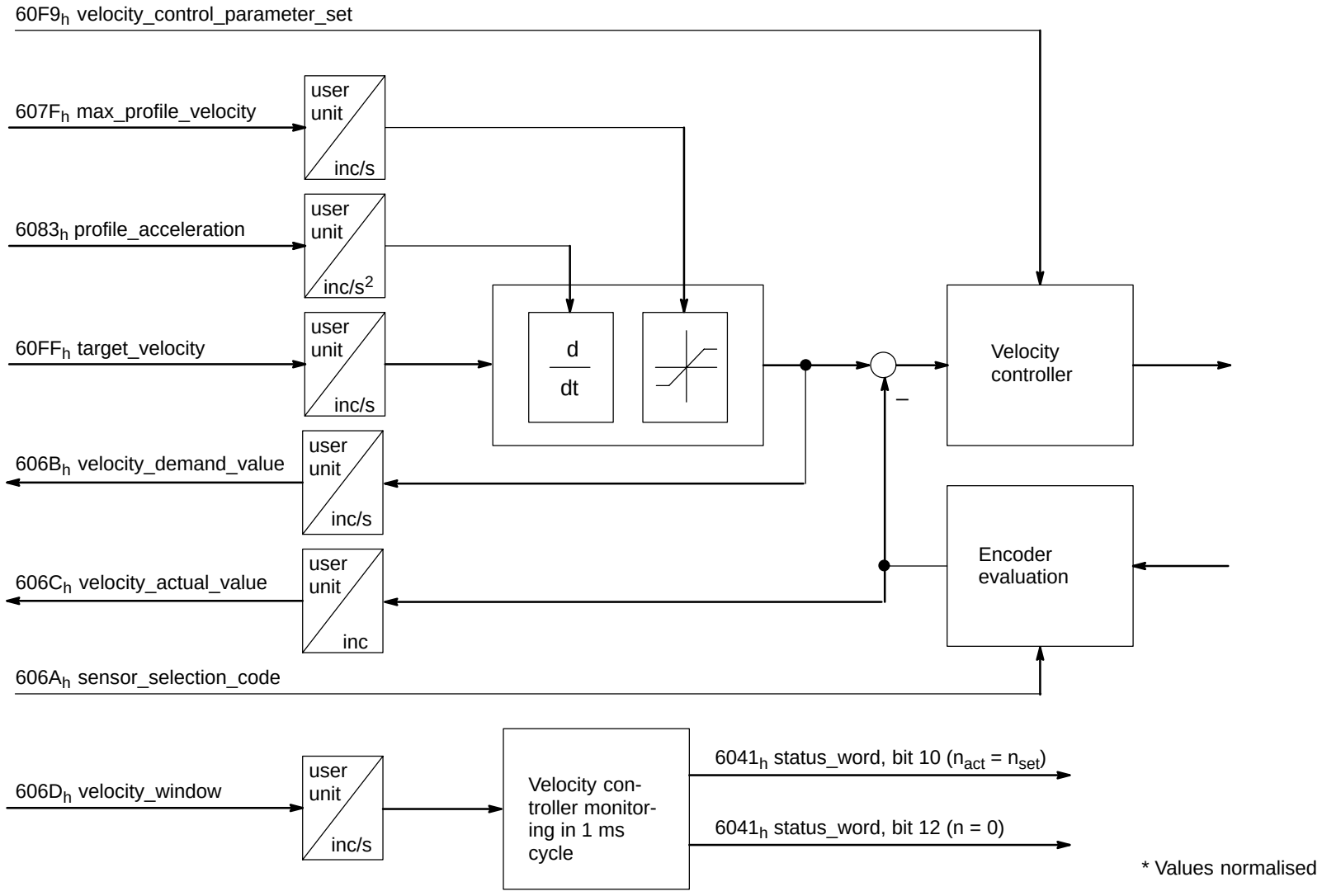
Input Data Description

Mode of Operation	Input parameters
Profile Velocity Mode	target_velocity, velocity_window, profile_acceleration, quick_stop_deceleration, quick_stop_option_code, motion_profile_type, max_profile_velocity, max_motor_speed

Output Data Description

Mode of Operation	Output parameters
Profile Velocity Mode	velocity_actual_value, velocity_demand_value, status_word

Profile Velocity Mode



12.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6
6071_h	VAR	target_torque	Integer16	13
6072_h	VAR	max_torque	Unsigned16	13
$607F_h$	VAR	max_profile_velocity	Unsigned32	8
6080_h	VAR	max_motor_speed	Unsigned32	8
6083_h	VAR	profile_acceleration	Unsigned32	8
6085_h	VAR	quick_stop_deceleration	Unsigned32	8
6086_h	VAR	motion_profile_type	Integer16	8

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
$400D_h$	VAR	actual_value_smoothing_integral_velocity_regulator	Unsigned16	rw
$606A_h$	VAR	sensor_selection_code	Integer16	rw
$606B_h$	VAR	velocity_demand_value	Integer32	ro
$606C_h$	VAR	velocity_actual_value	Integer32	ro
$606D_h$	VAR	velocity_window	Unsigned32	rw
$60FF_h$	VAR	target_velocity	Integer32	rw
$60F9_h$	RECORD	velocity_control_parameter_set		rw

position_encoder

The actual velocity can be obtained through differentiation from the position encoder and is given in position encoder increments.
It is described in greater detail in the position function.

C-400D$_h$ actual_value_smoothing_interval_velocity_regulator

This object determines whether the drive should establish a mean value from the actual value provided for internal processing in the velocity controller.

Index	**C-400D$_h$**
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: pv
Access	rw	
PDO mapping	no	
Units	0.1 µs	
Value range	625...10000	
Default value	initialised by drive	

C-606A$_h$ sensor_selection_code

This object determines the calculation of the actual speed via the position encoder.

Index	**C-606A$_h$**
Object code	VAR
Data type	Integer16

Value description

Object class	M: pv	O: –
Access	ro	
PDO mapping	yes	
Units	–	
Value range	0 = velocity actual value from position encoder	
Default value	0	

C-606B$_h$ velocity_demand_value

This object contains the output value of the interpolator, corrected by the output value of the position controller, as demand value for the velocity controller.

Index	**C-606B$_h$**
Object code	VAR
Data type	Integer32

Value description

Object class	M: pv	O: –
Access	ro	
PDO mapping	no	
Units	Velocity units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	$(-2^{31})..(2^{31}-1)$	
Default value	–	

C-606C$_h$ velocity_actual_value

This object contains the velocity actual value, the input value of the velocity controller.

Index	**C-606C$_h$**
Object code	VAR
Data type	Integer32

Value description

Object class	M: pv	O: –
Access	ro	
PDO mapping	yes	
Units	Velocity units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	$(-2^{31})..(2^{31}-1)$	
Default value	–	

C-606D$_h$ velocity_window

This object monitors whether the required velocity has been achieved after an acceleration or braking phase.

Index	C-606D$_h$
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: pv
Access	rw	
PDO mapping	no	
Units	Velocity units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	$0...2^{32}$	
Default value	weighting-dependent	

C-60FF$_h$ target_velocity

This object is the input value for the Setpoint Generator.

Index	C-60FF$_h$
Object code	VAR
Data type	Integer32

Value description

Object class	M: pv	O: –
Access	rw	
PDO mapping	yes	
Units	Velocity units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	$(-2^{31})..(2^{31}-1)$	
Default value	0	

C-60F9$_h$ velocity_control_parameter_set

This object controls the behaviour of the velocity controller. It contains two parameters for a PI control

Index	C-60F9$_h$
Object code	RECORD
Number of elements	2
Data type	Unsigned16

V: Gain

Sub-index	01$_h$	
Object class	M: –	O: hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	–	
Value range	0..65535	
Default value	Initialised by drive	

P-share (proportional gain) of the velocity controller. A preliminary setting is sufficient in many application cases. When an adaptation is necessary, the I-share in the transient response depends on the velocity actual value.

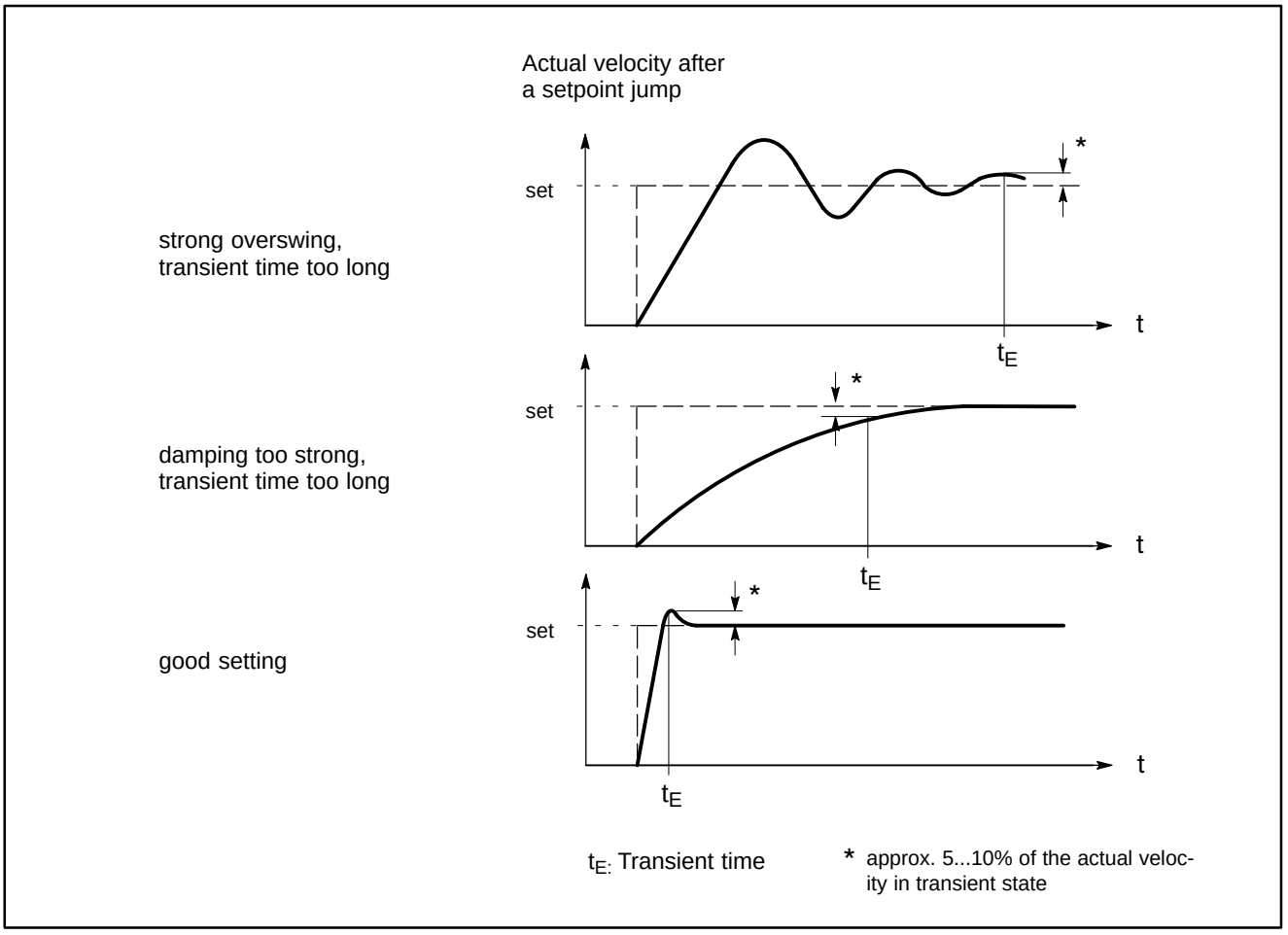
The aim of the adaptation is to minimise the transient time at a minimum overswing behaviour. If the controller behaviour has not been adapted optimally, the step response will have a strong overswing (related to the transient state: >10%) or will be dampened too much (see Fig. on page 12–8). This reduces the maximum possible drive dynamics and has a negative influence on contour definition.

The additional card OM-4 and an external storage oscilloscope are used to check the transient behaviour. The internal digital velocity actual values and velocity setpoints are converted to the corresponding output voltages by OM-4 and visualised using the storage oscilloscope.
Measurements should be made only with coupled axle mechanical system and an average load, if appropriate, in order to obtain practice-related data.

T_i: Integration time constant

Sub-index	02_h	
Object class	M: –	O: hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0..65535	
Default value	Initialised by drive	

I-share (integral action time) of the velocity controller. A preliminary setting is sufficient in many application cases. When an adaptation is necessary, the I-share in the transient response depends on the velocity actual value (see Figure).



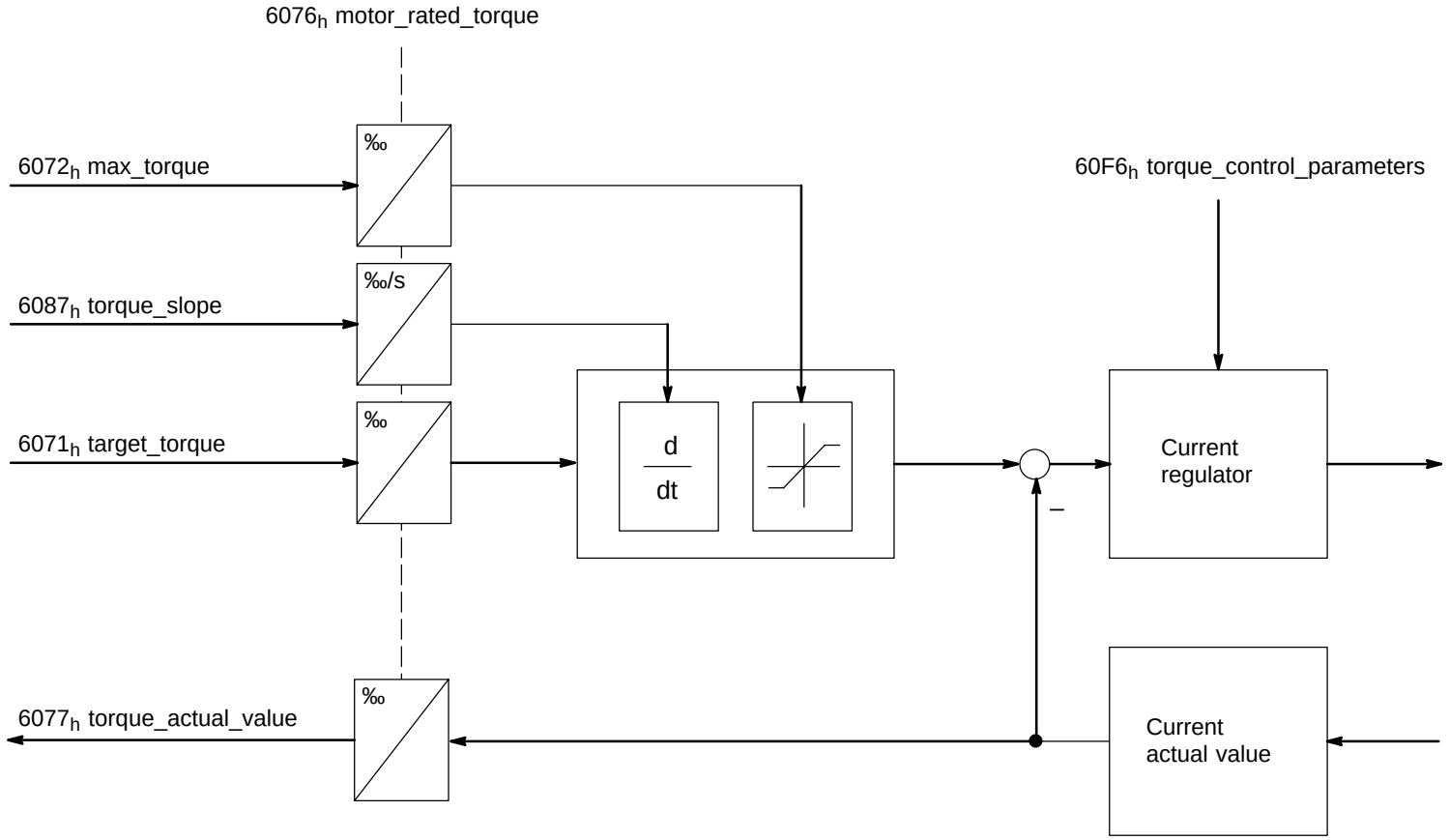
13 Profile Torque Mode

13.1 Overview

The Profile Torque Mode allows a host control system with velocity controller or transmission force controller to transmit the target_torque value, which is processed via the interpolator. The torque_slope and torque_profile_type parameters are required.

The same objects are used to operate **linear motors** in the Profile Torque Mode. All objects in connection with the ”torque” have to refer to a ”force” instead. As an example, the linear motor target force must be transmitted using the target_torque object. Refer to the object descriptions for additional information.

Profile Torque Mode



* Values normalised

13.2 Object Description

Objects defined in other Chapters

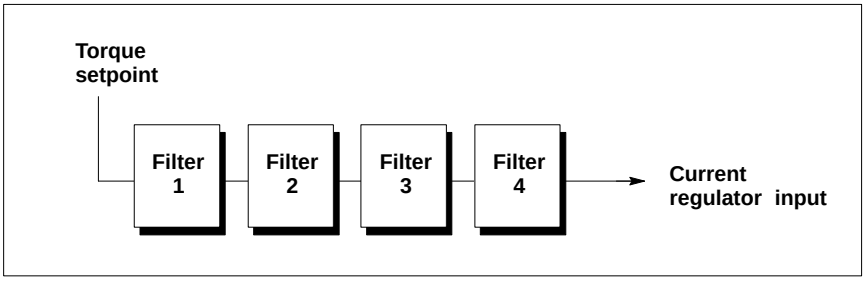
Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
4078_h	ARRAY	current_command_filter_filtertype	Unsigned16	rw
4079_h	ARRAY	current_command_filter_ limit_frequency_low_pass_filter	Unsigned16	rw
$407A_h$	ARRAY	current_command_filter_ quality_band_rejection_filter	Unsigned16	rw
$407B_h$	ARRAY	current_command_filter_center_ frequency_band_rejection_filter	Unsigned16	rw
6071_h	VAR	target_torque	Integer16	rw
6072_h	VAR	max_torque	Unsigned16	rw
6076_h	VAR	motor_rated_torque	Unsigned32	rw
6077_h	VAR	torque_actual_value	Integer16	ro
6087_h	VAR	torque_slope	Unsigned32	rw
6088_h	VAR	torque_profile_type	Integer16	rw
$60F6_h$	RECORD	torque_control_parameters		rw

C-4078$_h$ current_command_filter_filtertype

The drive has 4 series-connected 2nd order digital filters through which the torque setpoint is led to the current regulator input.
Each of the 4 filters can be parameterised as low-pass or band-rejection. It is also possible to completely switch off each filter.



In this manner, the torque slope of the drive may be accurately optimised to the respective requirements of your application.

Index	**C-4078$_h$**
Object code	ARRAY
Number of elements	4
Data type	Unsigned16

current_command_filter1_filtertype

Sub-index	**01$_h$**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = filter off 1 = low-pass 2 = band rejection	
Default value	0	

current_command_filter2_filtertype

Sub-index	**02$_h$**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = filter off 1 = low-pass 2 = band rejection	
Default value	0	

current_command_filter3_filtertype

Sub-index	**03_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = filter off 1 = low-pass 2 = band rejection	
Default value	0	

current_command_filter4_filtertype

Sub-index	**04_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = filter off 1 = low-pass 2 = band rejection	
Default value	0	

$C\text{-}4079_h$ current_command_filter_limit_frequency_low_pass_filter

Enter the limit frequency in Hz for those filters configured as low-pass in in $C\text{-}4078_h$. The signal range above the frequency entered will be dampened by –20 dB per decade.
The entered value will only be used if the corresponding filter was configured as low-pass.

Index	**$C\text{-}4079_h$**
Object code	ARRAY
Number of elements	4
Data type	Unsigned16

current_command_filter1_limit_frequency_low_pass_filter

Sub-index	**01_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter2_limit_frequency_low_pass_filter

Sub-index	**02_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter3_limit_frequency_low_pass_filter

Sub-index	**03_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter4_limit_frequency_low_pass_filter

Sub-index	**04_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

C-407A$_h$ current_command_filter_quality_band_rejection_filter

Enter the quality in C-407A$_h$ and the center frequency in Hz in C-407B$_h$ for those filters configured as band rejection in C-4078$_h$. The signal range at the frequency entered will be dampened by more than –40 dB.
You can determine the steepness of the filter slope by the quality. The higher the quality, the more narrow-band the suppressed frequency range.

Entered values will only be used if the corresponding filter was configured as band rejection.

Index	**C-407A$_h$**
Object code	ARRAY
Number of elements	4
Data type	Unsigned16

current_command_filter1_quality_band_rejection_filter

Sub-index	**01$_h$**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	1...1000	
Default value	1	

current_command_filter2_quality_band_rejection_filter

Sub-index	**02$_h$**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	1...1000	
Default value	1	

current_command_filter3_quality_band_rejection_filter

Sub-index	**03$_h$**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	1...1000	
Default value	1	

current_command_filter4_quality_band_rejection_filter

Sub-index	**04_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	–	
Value range	1...1000	
Default value	1	

C-407B$_h$ current_command_filter_center_frequency_band_rejection_filter

Enter the quality in C-407A$_h$ and the center frequency in Hz in C-407B$_h$ for those filters configured as band rejection in C-4078$_h$. The signal range at the frequency entered will be dampened by more than –40 dB.
You can determine the steepness of the filter slope by the quality. The higher the quality, the more narrow-band the suppressed frequency range.

Entered values will only be used if the corresponding filter was configured as band rejection.

Index	**C-407A$_h$**
Object code	ARRAY
Number of elements	4
Data type	Unsigned16

current_command_filter1_center_frequency_band_rejection_filter

Sub-index	**01_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter2_center_frequency_band_rejection_filter

Sub-index	**02_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter3_center_frequency_band_rejection_filter

Sub-index	**03_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

current_command_filter4_center_frequency_band_rejection_filter

Sub-index	**04_h**	
Object class	M: –	O: tq
Access	rw	
PDO mapping	no	
Units	Hz	
Value range	1...4000	
Default value	Initialised by drive	

C-6071$_h$ target_torque

This object is the input value for the torque controller in Profile Torque Mode.

Index	**C-6071$_h$**
Object code	VAR
Data type	Integer16

Value description

Object class	M: tq	O: –
Access	rw	
PDO mapping	yes	
Units	per thousand of rated torque	
Value range	0...4000	
Default value	0	

C-6072$_h$ max_torque

Maximum permissible torque in the motor.

Index	C-6072$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: tq, hc, pv, pp, ip
Access	rw	
PDO mapping	yes	
Units	per thousand of rated torque	
Value range	10...4000	
Default value	3000	

C-6076$_h$ motor_rated_torque

Rated torque of the motor. The value is read from the electronic nameplate of the motor. All relative torque data refer to this value.
For linear motors, this object contains the motor rated torque in 0.001 N.

Index	C-6076$_h$
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: –	O: tq, hc, pv, pp, ip
Access	ro	
PDO mapping	no	
Units	0.001 Nm	
Value range	0..(2^{32} –1)	
Default value	motor-dependent	

C-6077$_h$ torque_actual_value

This object contains the torque actual value in the motor.

Index	**C-6077$_h$**
Object code	VAR
Data type	Integer16

Value description

Object class	M: –	O: tq, hc, pv, pp, ip
Access	ro	
PDO mapping	yes	
Units	per thousand of rated torque	
Value range	–32768..32767	
Default value	0	

C-6087$_h$ torque_slope

This object describes the rate of change of the torque.

Index	**C-6087$_h$**
Object code	VAR
Data type	Unsigned32

Value description

Object class	M: tq	O: –
Access	rw	
PDO mapping	yes	
Units	per thousand of rated torque per ms	
Value range	0..(2^{32} –1)	
Default value	max. ramp	

C-6088$_h$ torque_profile_type

This object establishes a linear ramp for torque changes.

Index	**C-6088$_h$**
Object code	VAR
Data type	Integer16

Value description

Object class	M: tq	O:–
Access	ro	
PDO mapping	optional	
Units	none	
Value range	0 = linear ramp, trapezoidal profile	
Default value	0	

C-60F6$_h$ torque_control_parameters

This object contains the parameter for the current regulator.

Index	**C-60F6$_h$**
Object code	RECORD
Number of elements	4
Data type	Unsigned16

proportional_gain_1_current_regulator

P-share of the current regulator for the torque-generating current portion. As a rule, the preset value remains unchanged.

Sub-index	**01$_h$**	
Object class	M: –	O: tq, hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	–	
Value range	0...65535	
Default value	Initialised by drive	

integral_action_time_1_current_regulator

I-share of the current regulator for the torque-generating current portion. As a rule, the preset value remains unchanged.

Sub-index	**02_h**	
Object class	M: –	O: tq, hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...65535	
Default value	Initialised by drive	

proportional_gain_2_current_regulator

P-share of the current regulator for the flux-generating current portion. As a rule, the preset value remains unchanged.

Sub-index	**03_h**	
Object class	M: –	O: tq, hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	–	
Value range	0...65535	
Default value	Initialised by drive	

integral_action_time_2_current_regulator

I-share of the current regulator for the flux-generating current portion. As a rule, the preset value remains unchanged.

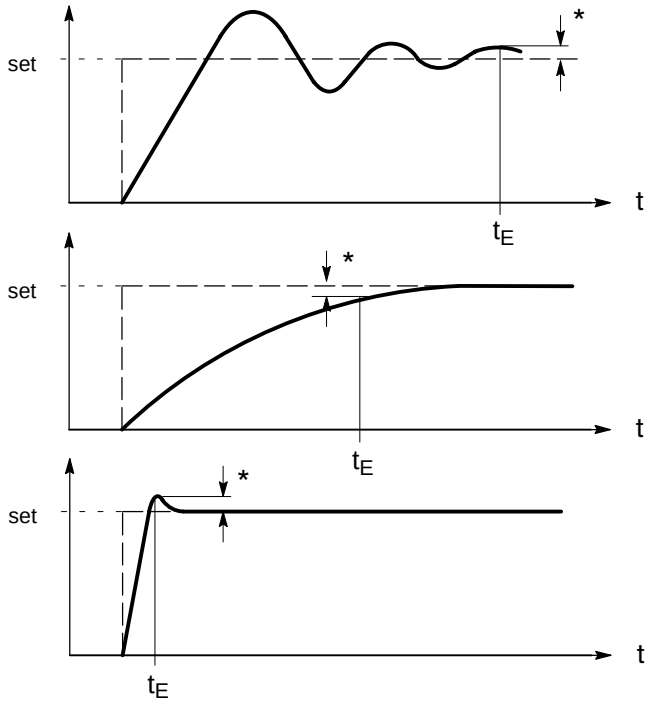
Sub-index	**04_h**	
Object class	M: –	O: tq, hc, pv, pp, ip, vl
Access	rw	
PDO mapping	no	
Units	ms	
Value range	0...65535	
Default value	Initialised by drive	

Output current after
a setpoint jump

strong overswing,
transient time too long

damping too strong,
transient time too long

good setting



t_E: Transient time

* approx. 5...10% of the output current in transient state

14 Velocity Mode

14.1 Overview

The Velocity Mode is composed of the following subfunctions:

- Reference Calculation
- Factor Function, Reverse Factor Function
- Velocity Limit Function
- Velocity Motor Limit Function
- Ramp Function
- Ramp Min Function
- Closed / Open Loop Control Function

All drive systems, including frequency inverters, can be operated in Velocity Mode. The unit in which the velocity values in the speed functions are calculated is rpm.
Most applications use a velocity setpoint and a control_word for switching the drive on and off.
See Device Control in chapter 6.

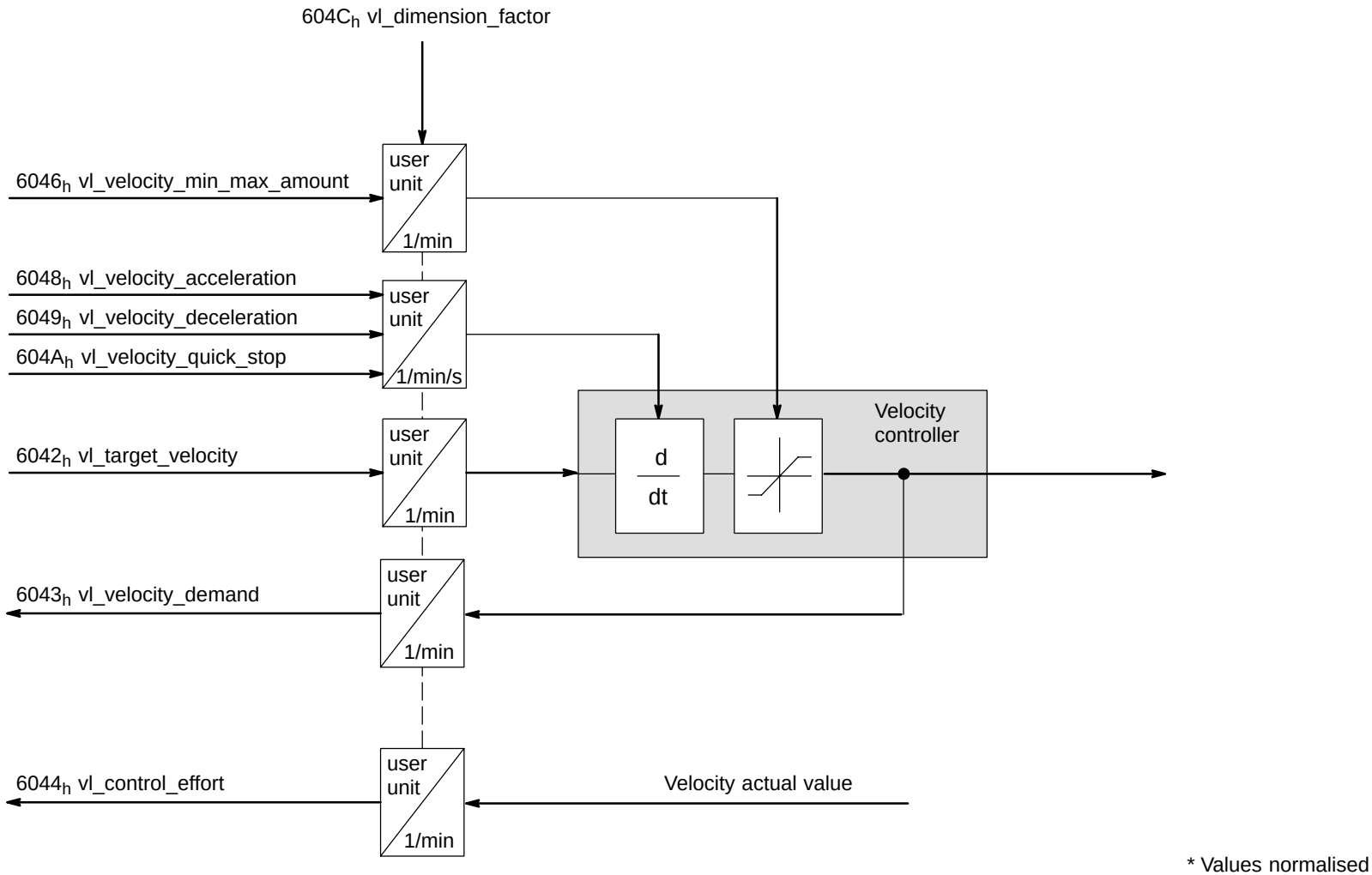
Input Data Description

Mode of Operation	Input Parameters Used
Velocity Mode	vl_target_velocity, vl_dimension_factor, velocity_min_max_amount, vl_velocity_acceleration, vl_velocity_deceleration, vl_velocity_quick_stop

Output Data Description

Mode of Operation	Output Parameters used
Velocity Mode	vl_control_effort, vl_velocity_demand

Velocity Mode



14.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
6042_h	VAR	vl_target_velocity	Integer16	rw
6043_h	VAR	vl_velocity_demand	Integer16	ro
6044_h	VAR	vl_control_effort	Integer16	ro
6046_h	ARRAY	vl_velocity_min_max_amount	Unsigned32	rw
6047_h	ARRAY	vl_velocity_min_max	Unsigned32	rw
6048_h	RECORD	vl_velocity_acceleration	ramp	rw
6049_h	RECORD	vl_velocity_deceleration	ramp	rw
$604C_h$	ARRAY	vl_dimension_factor	Integer32	rw

C-6042_h vl_target_velocity

This object specifies the velocity.
The vl_target_velocity is converted to the unit [rpm] by multiplying the vl_target_velocity by the vl_dimension_factor.

Index	C-6042_h
Object code	VAR
Data type	Integer16

Value description

Object class	M:vl	O: –
Access	rw	
PDO mapping	yes	
Units	rpm * vl_dimension_factor	
Value range	–32768..32767	
Default value	0	

C-6043$_h$ vl_velocity_demand

The currently specified velocity, scaled to the unit of vl_target_velocity (drive internal value).

Index	**C-6043$_h$**
Object code	VAR
Data type	Integer16

Value description

Object class	M:vl	O: –
Access	ro	
PDO mapping	yes	
Units	rpm * vl_dimension_factor	
Value range	–32768..32767	
Default value	– (output variable of drive)	

C-604C$_h$ vl_dimension_factor

This object serves to include gearing in the calculations or to scale the frequencies and specific units of the application. It influences the specified setpoint, the velocity limit and the ramp_function as well as the output variables of the speed function.

Calculating the vl_dimension_factor

Every speed consists of a specific unit referred to a unit of time (e.g. 1/sec, bottles/min, m/sec,...).
The purpose of the dimension factor is to convert this specific unit to the revolutions/minute unit.

Specific unit • DF = 1 revolution (motor shaft)

I • DF = O

where:

I = vl_target_velocity as input value, in application specific units

O = output value as speed value in [rpm]

DF = vl_dimension_factor

$$\text{Unit: } [DF] = \frac{1}{\text{Specific unit}} \bullet \frac{1}{\text{min}}$$

Index	**C-604C$_h$**
Object code	ARRAY
Number of elements	2
Data type	Integer32

vl_dimension_factor_numerator

Sub-index	**01_h**	
Object class	M: –	O:vl
Access	rw	
PDO mapping	no	
Units	–	
Value range	$(-2^{31})..(2^{31}-1)$, 0 not permissible	
Default value	1	

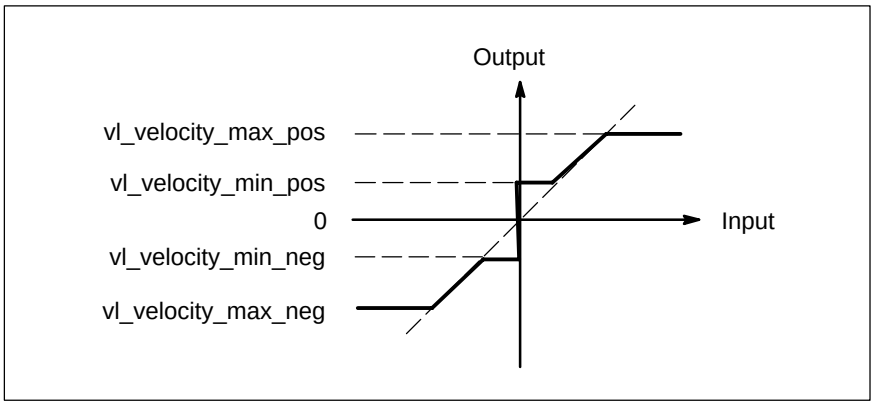
vl_dimension_factor_denominator

Sub-index	**02_h**	
Object class	M: –	O:vl
Access	rw	
PDO mapping	no	
Units	–	
Value range	$(-2^{31})..(2^{31}-1)$, 0 not permissible	
Default value	1	

C-6046$_h$ vl_velocity_min_max_amount

The sub-index vl_velocity_max_amount is mapped internally by the vl_velocity_max_pos and vl_velocity_max_neg values.
Only the positive values are returned if the object vl_velocity_min_max_amount is read out.

Transfer characteristic from vl_velocity_min_max_amount.



Index	**C-6046$_h$**
Object code	ARRAY
Number of elements	2
Data type	Unsigned32

vl_velocity_min_amount

Sub-index	**01_h**	
Object class	M: vl	O: –
Access	ro	
PDO mapping	no	
Units	rpm * vl_dimension_factor	
Value range	0	
Default value	0	

vl_velocity_max_amount

Sub-index	**02_h**	
Object class	M: vl	O:–
Access	rw	
PDO mapping	no	
Units	rpm * vl_dimension_factor	
Value range	$0..(2^{32} –1)$	
Default value	max. motor speed	

C-6048$_h$ vl_velocity_acceleration

This object specifies the slope of the acceleration ramp.

$$\text{vl\_velocity\_acceleration} = \frac{\text{delta\_speed}}{\text{delta\_time}} = a_B$$

Index	**C-6048$_h$**
Object code	RECORD
Number of elements	2

delta_speed

Sub-index	**01$_h$**	
Object class	M: vl	O: –
Access	rw	
PDO mapping	no	
Units	rpm * vl_dimension_factor	
Value range	0..(2^{32} –1)	
Default value	weighting-dependent	
Data type	Unsigned32	

delta_time

Sub-index	**02$_h$**	
Object class	M: vl	O:–
Access	rw	
PDO mapping	no	
Units	seconds (s)	
Value range	0...65535	
Default value	1	
Data type	Unsigned16	

C-6049$_h$ vl_velocity_deceleration (= C-6048$_h$)

This object specifies the slope of the deceleration ramp.

$$\text{vl\_velocity\_deceleration} = \frac{\text{delta\_speed}}{\text{delta\_time}} = a_V$$

Index	C-6049$_h$
Object code	RECORD
Number of elements	2

delta_speed

Sub-index	01$_h$	
Object class	M: vl	O: –
Access	rw	
PDO mapping	no	
Units	rpm * vl_dimension_factor	
Value range	0..(2^{32} –1)	
Default value	weighting-dependent	
Data type	Unsigned32	

delta_time

Sub-index	02$_h$	
Object class	M: vl	O:–
Access	rw	
PDO mapping	no	
Units	seconds (s)	
Value range	0...65535	
Default value	1	
Data type	Unsigned16	

C-604A$_h$ vl_velocity_quick_stop

This object establishes a ramp for quick stop.

$$\text{velocity\_quick\_stop} = \frac{\text{delta\_speed}}{\text{delta\_time}} = a_S$$

Index	**C-604A$_h$**
Object code	RECORD
Number of elements	2

delta_speed

Sub-index	**01$_h$**	
Object class	M: –	O: vl
Access	rw	
PDO mapping	no	
Units	rpm * vl_dimension_factor	
Value range	0..(2^{32} –1)	
Default value	weighting-dependent	
Data type	Unsigned32	

delta_time

Sub-index	**02$_h$**	
Object class	M: –	O:vl
Access	rw	
PDO mapping	yes	
Units	seconds (s)	
Value range	0...65535	
Default value	1	
Data type	Unsigned16	

C-6044$_h$ vl_control_effort

This object contains the velocity at the motor or load, scaled to the unit of the vl_target_velocity. Depending on the system, velocity deviations may occur between the vl_control_effort and the physical velocity.

Index	**C-6044$_h$**
Object code	VAR
Data type	Integer16

Value description

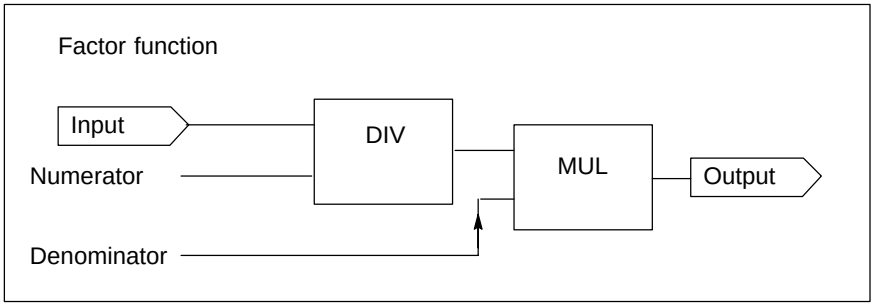
Object class	M: vl	O:–
Access	ro	
PDO mapping	yes	
Units	rpm * vl_dimension_factor	
Value range	–32768..32767	
Default value	– (output variable of drive)	

14.3 Functional Description

14.3.1 Factor Function and Reverse Factor Function

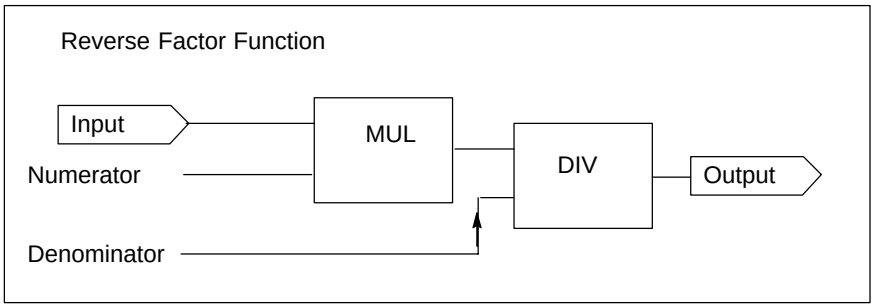
The Factor Function multiplies the input variables by the assigned factors:

- The vl_target_velocity is multiplied by the vl_dimension_factor.
- The values of the velocity limit and the values for the Ramp Function are multiplied by the vl_dimension_factor.



The Reverse Factor Function divides the input variables by the assigned factors.

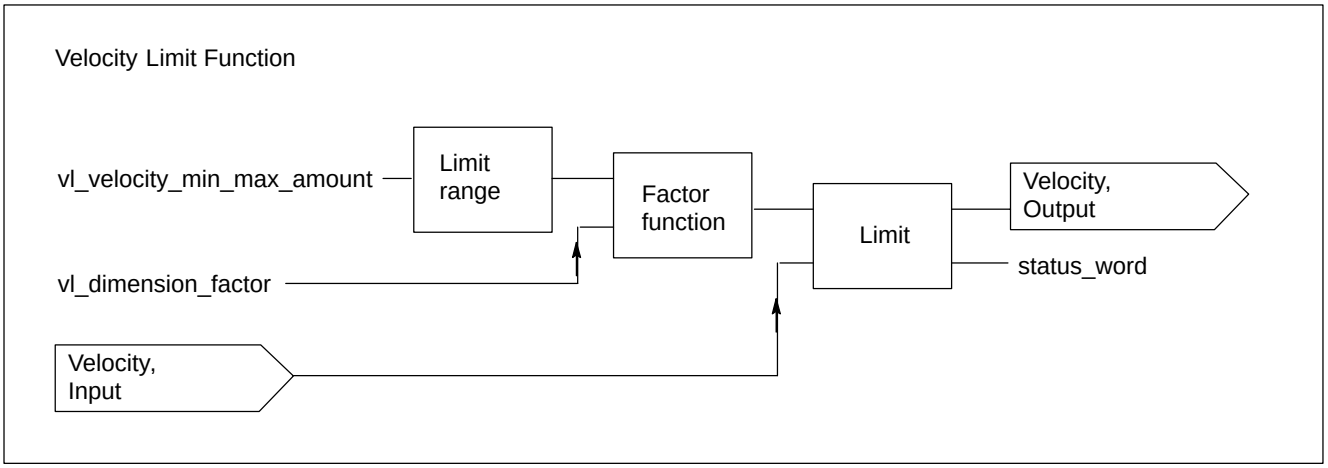
- The output variables of the Velocity Mode are calculated by division with the vl_dimension_factor and therefore returned to the scaling of the specified setpoint.



14.3.2 Velocity Limit Function

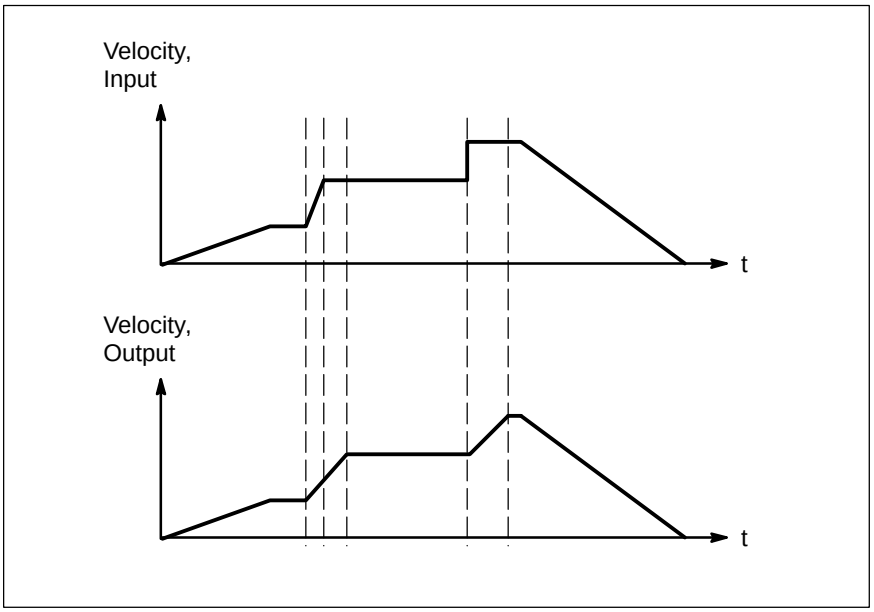
The velocity limit defines the valid velocity range for the drive. Limits can be specified in the user specific units by including the vl_dimension_factor in the speed limit.

The limit value message is generated if the input value is outside of the range of the velocity limit. The limit value message is mapped as one bit in the status_word.



14.3.3 Ramp Function

The ramp function increases and decreases the velocity. The velocity output value is equal to the input value as long as the changes are below a_B, a_V, or a_S.



15 Motion Control Mode

15.1 Overview

Motion Control Mode is composed of the program management and the actual MC functionality. In connection with CANopen, the smallest cycle time of the position setpoint generator is 4 ms.

A MC program consists of a maximum of 32 travel blocks. The block information position, speed, acceleration, deceleration and block control_word are transmitted by one object each. Individual block information may be exchanged during operation, changes becoming effective the next time the block is selected. If the speed or acceleration values are missing for individual positions, the values from objects C-6081 and C-6083 are used.

A program always includes the data:

- Position weighting C-6089 and C-608A,
- Speed weighting C-608B and C-608C and
- Acceleration weighting C-608D and C-608E.

Thus it is ensured that the program data will be interpreted correctly in the drive.

Input Data Description

Mode of Operation	Input Parameters used
Motion Control Mode	mc_control_word, mc_block_selection, parameter_roundaxis mc_target_positions, mc_positioning_velocity, mc_positioning_accelerations, mc_positioning_decelerations, mc_positioning_controls

Output Data Description

Mode of Operation	Output Parameters used
Motion Control Mode	velocity_demand_value, mc_status_word, mc_error_list

15.2 Object Description

Objects defined in other Chapters

Index	Code	Name	Type	Chapter
$606B_h$	VAR	velocity_demand_value	Integer32	12
6081_h	VAR	profile_velocity	Unsigned32	8
6083_h	VAR	profile_acceleration	Unsigned32	8
6040_h	VAR	control_word	Unsigned16	6
6041_h	VAR	status_word	Unsigned16	6
6067_h	VAR	position_window	Unsigned32	10

Objects defined in this Chapter

Index	Code	Name	Type	Attr.
4899_h	VAR	mc_control_word	Unsigned16	rw
$489A_h$	VAR	mc_status_word	Unsigned16	ro
$489B_h$	VAR	mc_block_selection	Unsigned16	rw
$489E_h$	ARRAY	mc_error_list	Unsigned16	ro
$48A2_h$	VAR	parameter_roundaxis	Unsigned16	rw
$48AC_h$	ARRAY	mc_target_positions	Integer32	rw
$48B6_h$	ARRAY	mc_positioning_velocity	Integer32	rw
$48C0_h$	ARRAY	mc_positioning_acceleration	Integer32	rw
$48CA_h$	ARRAY	mc_positioning_deceleration	Integer32	rw

C-4899$_h$ mc_control_word

This object controls the MC functionality.

Index	C-4899$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: mc
Access	rw	
PDO mapping	yes	
Units	–	
Value range	0...15	
Default value	0	

Data description

Bit	Name
0	Select mode of operation
1	
2	Reference cams
3	Basic position
4	Start / stop

Bit 0, 1

Bit 1	Bit 0	Mode of Operation
0	0	Automatic mode, position table is processed.
0	1	Homing
1	0	Jog negative (setting up mode)
1	0	Jog positive (setting up mode)

Bit 2 Is set when the reference cam has been reached.

Bit 3 With ”Basic Position”, an active or interrupted travel movement is cleared and the output signal ”In Position” is activated. The signal is effective in the automatic mode as well as in the setting up mode.

Bit 4 A positive edge in ”start/stop” starts a travel motion.

C-489A$_h$ mc_status_word

The status_word indicates the current state of the drive.

Index	C-489A$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: mc
Access	ro	
PDO mapping	yes	
Units	–	
Value range	–	
Default value	–	

Data description

Bit	Name
0	MC active (ready for operation)
1	Drive is referenced (InRef)
2	Axle active
3	In position
4	End of block
5	Basic position
6	MC error status
7	Start edge detected

Bit 0: MC active.

Is set when the MC function is ready for operation.

Bit 1: Drive is referenced.

Drives with incremental encoder system have to be referenced for automatic operation. Only then can blocks travel in automatic mode.
This is not required in case of drives with absolute encoder systems. Bit 1 is always set parallel to bit 0.

☞ **Moving the axle in jog mode is also possible without referenced drive!**

Bit 2: Axle active.

Is set as soon as the axle carries out a travel motion.
Is reset when

- the home position has been reached in the Homing Mode.
- the axle has come to a standstill in the jog mode of operation
- the end of the block has been reached in automatic mode (bit 4=1)
- basic position has been triggered.

Bit 3: In position.

Is set as soon as the current position is sited within the position window (C-6067) with active controller release and no encoder error is present. In case of a remaining path that is greater than the position window or after interruption of the travel movement (C-4899, bit 4: 1 → 0) ”in position” remains=0 and ”axle active”=1 to signalise the interruption. ”Basic position” at this point in time clears the existing distance to go and sets ”in position” to 1.

☞ **Without controller release, bit 3 is always set.**

Bit 4: End of block.

Is set as soon as the travel block which started in Automatic Mode is completed.
Is reset by a negative edge of bit 4 in C-4899.

Bit 5: Basic position.

Is set as soon as the basic position (bit 3 in C-4899) has been correctly executed internally and new specifications can be made.

Bit 6: MC error status.

Is set when the drive detects an MC fault. The cause of the fault may be read out via object C-489E.
It is reset by basic position (bit 3 in C-4899).

Bit 7: Start edge detected.

Is set as soon as a positive edge of the ”start/stop” signal (C-4899, bit 4: 0 → 1) has been detected.
A further write process at C-4899 resets ”start edge detected”.

C-489B$_h$ mc_block_selection

In automatic mode, this object is used to select one out of 32 table blocks from the current position table.
The axle movement is started with the next positive edge of the start/stop signal (C-4899, bit 4: 0 → 1).

Index	**C-489B$_h$**
Object code	VAR
Data type	Unsigned16

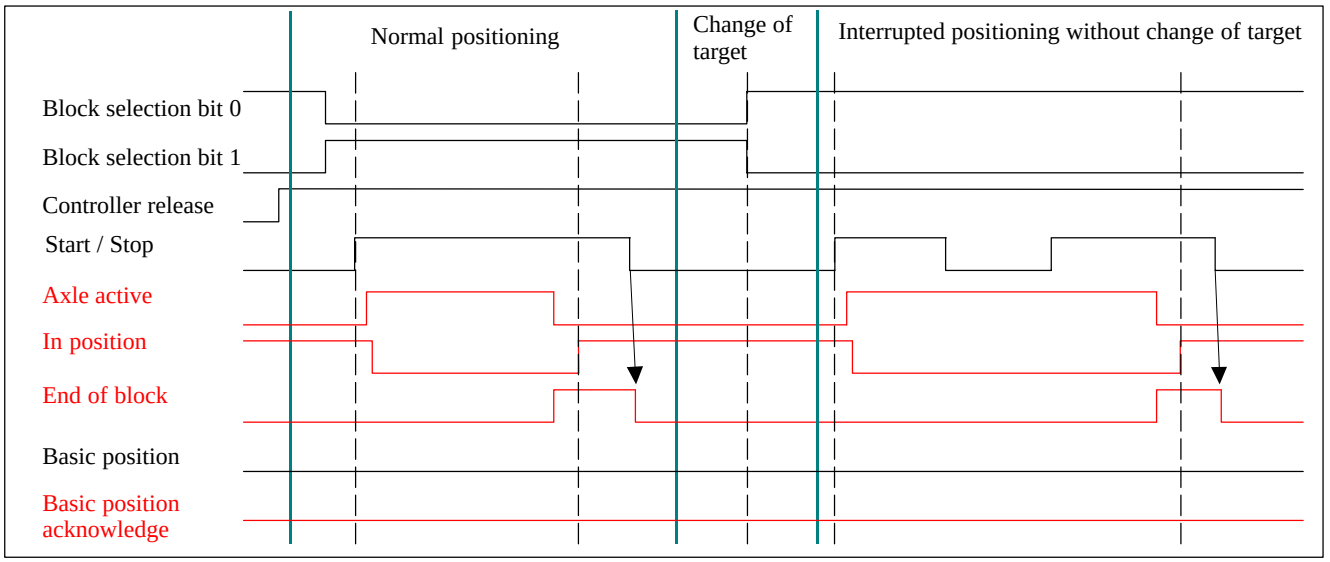
Value description

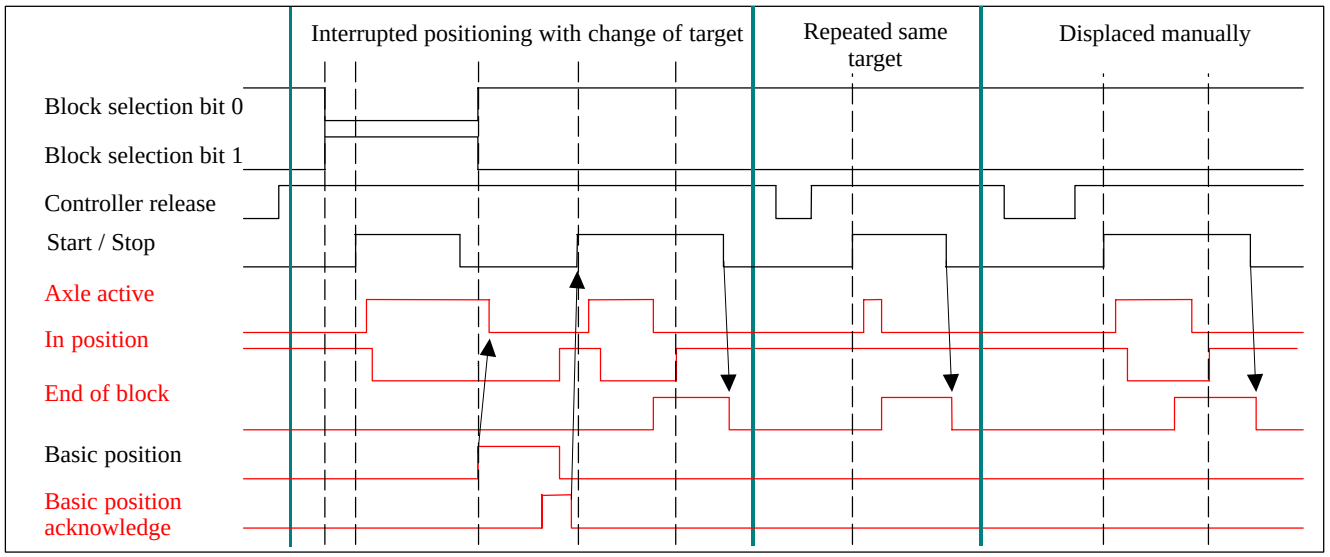
Object class	M: –	O: mc
Access	rw	
PDO mapping	yes	
Units	–	
Value range	0...31	
Default value	0	

Data description

Bit	**Name**
0	Block selection 0...31
1	
2	
3	
4	

Typical Signal Sequences for Control and Status Bits in Automatic Mode





C-489E$_h$ mc_error_list

Contains numbers, warnings or faults in connection with Motion Control which do not require immediate shutdown of the axles. Drive operation is disturbed, however.

The numbers in C-489E are cleared by:

- basic position **or**
- new running up (RESET) of the drive

Index	**C-489E$_h$**
Object code	ARRAY
Number of elements	0...8
Data type	Unsigned16

mc_error_1 ... mc_error_8

Object class	M: –	O: mc
Access	ro	
PDO mapping	no	
Units	–	
Value range	–	
Default value	–	

Data description

Error number	**Meaning**	**Action**
System:		
102	Overflow of error list. More errors have occurred than can be recorded by the error list.	–
Interpolator:		
302	Interpolator not ready for operation. Interpolator not completely initialised or not ready for operation due to a fault.	Run system up again.
305	Internal interpolator fault.	Run system up again.
Block control:		
402	No program loaded or selected.	Select program.
406	Axle not in home position, prerequisite for automatic operation.	Homing of axle.
408	Selected table block is not included in the program.	Load program, select block.

C-48A2$_h$ parameter_roundaxis

For endless axles (rotary axles with activated modulo calculation): With C-2067.01 = “Modulo value” and C-2067.02 = 1 ≙ modulo active) you may determine whether the target position should be reached:

- in a ”uniform sense of rotation” or
- on the ”optimum path”.

Index	C-48A2$_h$
Object code	VAR
Data type	Unsigned16

Value description

Object class	M: –	O: mc
Access	rw	
PDO mapping	possible	
Units	–	
Value range	0...3	
Default value	1	

Data description

Bit	Name
0	0 = uniform sense of rotation: negative direction 1 = uniform sense of rotation: positive direction
1	0 = approach in the ”uniform sense of rotation”, according to bit 0 1 = approach on ”optimum path”

Bit 0: Uniform sense of rotation

In case of the **”uniform sense of rotation”**, all positions are always approached in the direction determined by bit 0.

Prerequisites:
– active modulo calculation
– an absolute position has been programmed
– limit switches not active

Bit 1: Approach on optimum path

In case of **”approach on the optimum path”**, the target position is approached on the shortest path.

Prerequisites:
– active modulo calculation
– an absolute position has been programmed
– limit switches not active
– ”homing” is not active
– programmed position within the modulo value

☞ **Position specifications outside of the current modulo value C-2067.01 are first transformed into a corresponding position within the modulo range. The transformed position is then principally approached in a uniform sense of rotation and not on the optimum path.**

C-48AC$_h$ mc_target_position

This object is part of the CAN-MC program. It contains up to 32 target positions for the MC functionality.
The block to be travelled is selected with C-489B and the target position is approached taking into account:

- mc_positioning_velocity (C-48B6)
- mc_positioning_acceleration (C-48C0)
- mc_positioning_deceleration (C-48CA)

☞ **Depending on C-48D4, the value given here will be interpreted as absolute or incremental position.**

Index	**C-48AC$_h$**
Object code	ARRAY
Number of elements	0...32
Data type	Integer32

mc_target_position_1 ... mc_target_position_32

Object class	M: –	O: mc
Access	rw	
PDO mapping	no	
Units	Position units, as in position_notation_index (C-6089) and position_dimension_index (C-608A)	
Value range	-2^{31} ... (2^{31}–1)	
Default value	–	

C-48B6$_h$ mc_positioning_velocity

This object is part of the CAN-MC program. It contains velocities used to approach the target positions. If the velocity value is missing for one or several target positions, the object C-6081 will be used as substitute value.

Index	**C-48B6$_h$**
Object code	ARRAY
Number of elements	0...32
Data type	Integer32

mc_positioning_velocity_1 ... mc_positioning_velocity_32

Object class	M: –	O: mc
Access	rw	
PDO mapping	no	
Units	Velocity units, as in velocity_notation_index (C-608B) and velocity_dimension_index (C-608C)	
Value range	-2^{31} ... (2^{31}–1)	
Default value	–	
Substitute value	C-6081$_h$	

C-48C0$_h$ mc_positioning_acceleration

This object is part of the CAN-MC program. It contains values used to accelerate to the positioning speed. If the acceleration value is missing for one or several target positions, the object C-6083 will be used as substitute value.

Index	**C-48C0$_h$**
Object code	ARRAY
Number of elements	0...32
Data type	Integer32

mc_positioning_acceleration_1 ... mc_positioning_acceleration_32

Object class	M: –	O: mc
Access	rw	
PDO mapping	no	
Units	Acceleration units, as in acceleration_notation_index (C-608D) and acceleration_dimension_index (C-608E)	
Value range	-2^{31} ... (2^{31}–1)	
Default value	–	
Substitute value	C-6083$_h$	

C-48CA$_h$ mc_positioning_deceleration

This object is part of the CAN-MC program. It contains values used to decelerate at the end of the block. If the deceleration value is missing for one or several target positions, the object C-6083 will be used as substitute value.

Index	**C-48CA$_h$**
Object code	ARRAY
Number of elements	0...32
Data type	Integer32

mc_positioning_deceleration_1 ... mc_positioning_deceleration_32

Object class	M: –	O: mc
Access	rw	
PDO mapping	no	
Units	Acceleration units, as in acceleration_notation_index (C-608D) and acceleration_dimension_index (C-608E)	
Value range	-2^{31} ... (2^{31}–1)	
Default value	–	
Substitute value	C-6083$_h$	

C-48D4$_h$ mc_positioning_controls

This object is part of the CAN-MC program. It contains information on whether the target position entered in C-48AC is to be approached absolutely or incrementally.

Index	**C-48CA$_h$**
Object code	ARRAY
Number of elements	0...32
Data type	Unsigned16

mc_positioning_control_1 ... mc_positioning_control_32

Object class	M: –	O: mc
Access	rw	
PDO mapping	no	
Units	–	
Value range	0 = target position is absolute 1 = target position is incremental	
Default value	–	
Substitute value	C-6083$_h$	

16 Emergency Messages

16.1 Structure

Emergency messages are assigned the highest possible priority to ensure that they get access to the bus without latency. By default, the emergency messages contain an error field with pre-defined error numbers and additional information.

- the two lower bytes (Low Bytes) contain the error code Unsigned32
- the two higher bytes (High Bytes) contain the error classification and additional information

Error numbers from $xx00_h$ to $xx7F_h$ are defined in the communication profile DS-301 or in the drive profile DSP-402. Error numbers between $xx80_h$ and $xxFF_h$ can be used manufacturer specific.

16.2 Error Code Meanings

Error Code (Hex)	Error indication	Meaning	Action
0000	–	**Operation**	–
2000	F04	**Overcurrent** Admissible peak current exceeded	• Reduce load • Extend slow down, acceleration ramps • Check controller optimisation • Check motor • Check application
3000	F30	**Overvoltage in DC link**	• Reduce regenerating capacity (prolong deceleration time, reduce braking current) • In case of Servodyn-D, pay attention to emergency message on VM
4000	F08	**Overtemperature motor** Motor winding > 145°C, Temperature sensor or line defective (–273°C)	• Reduce motor load, use bigger motor • Check cooling system • Check encoder line • Check operating conditions
4310	F07	**Overtemperature module** Heat sink temperature > 80°C	• Increase air supply • Check fan
5100	F35	**External fault of 24 VDC** 24 V module supply not within admissible range	• Check 24 VDC supply • In case of Servodyn-D, check fuse F1 in VM
7300	F60	**Initialisation error I^2C bus**	• Exchange module
7320	F70	**Encoder error** Encoder sends no absolute position	• Check encoder connection • Exchange motor/encoder

Error Code (Hex)	Error indication	Meaning	Action
8100	F10	**Communication error CANopen**	• Check CAN line • Adjust SYNC time of all modules
8130	F10	**Life Guard Error**	• Check CAN line • Check if Master is still active • Check Node Guarding times
8400	F15	**Maximum speed exceeded** The speed limit C-6410.08$_h$ has been exceeded.	• Observe speed limit
8500	F14	**Position limit value exceeded** The specified setpoint results in an axle position outside of the travelling range.	• Check position limit parameter C-607D$_h$ • Check setting of software limit switch
8611	F13	**Excessive controller deviation** Drive was not able to follow the specified setpoints.	• Compare maximum setpoint specifications with speed limit value (C-6410.08$_h$) • Check lag
E015	F97	**VM error** Torque switched off due to a failure (e.g. overvoltage/undervoltage)	• Check emergency message at VM • Clear fault using ”Reset Fault”
E016	F98	**VM switch off** Shutdown of motors with active control system due to a switch off command	• Switch on VM • Check EMERGENCY STOP • Clear fault using ”Reset Fault”
E017	F06	**Real time error** Watchdog error	• Restart module • Exchange module
E01E	F96	**Plausibility error velocity (speed) controller**	• Check motor cable • Motor is blocked • Switch off torque reduction • Check Servodyn-D: check DC link voltage, check DC link connection between the modules, if applicable (in case of DM..A,B,C in the backplane modules). To do this, switch off the modules because of **dangerous** electrical voltage!
E01F	F80	**Motion Control error**	• For detailed information see C-489E, page 15–8
E020	F69	**Wrong temperature sensor motor**	• Install matching motor
E021	F69	**Timeout temperature measurement**	• Switch off digital/analog conversion for analog outputs on optional module OM04 (DAC channels)
E022	F06	**Real time error** CPU computing time insufficient	• Reduce computer load due to DSS-D (e.g. switch off oscilloscope) • Exchange module
E033	F71	**Driver identifier invalid**	• Exchange module
E034	F61	**Initialisation error AD converter**	• Check motor cable • Exchange module
E038	F72	**Wrong motor – module allocation** Motor encoder and encoder interface at the inverter do not match.	• Install matching components
E03E	F53	**Error in writing parameters**	
E03F	F54	**Error in reading parameters**	

Error Code (Hex)	Error indication	Meaning	Action
E041	F70	**Encoder error** Parity error of the absolute position	
E042	F70	**Encoder error** Error in saving the zero position	
E043	F70	**Encoder error** Timeout when reading via SSI interface	
E044	F70	**Encoder error** Check sum error during data reception	
E045	F70	**Encoder error** Timeout when writing via SSI interface	
E046	F70	**Encoder error** PPR count not defined	
E047	F70	**Encoder error** Error when initialising the I^2C bus	
E048	F70	**Encoder error** Error when reading via I^2C bus	
E049	F70	**Encoder error** Wrong PPR count	
E04A	F70	**Encoder error** Encoder parameter value is zero	
E04B	F70	**Encoder error** Timeout when reading via EnDat interface	
E04C	F70	**Encoder error** No encoder interface	
E04D	F70	**Encoder error** Data transmission disturbed	
E04E	F70	**Encoder error** Malfunction EnDat measuring system	
E04F	F70	**Encoder error** No encoder	

Notes:

A Appendix

A.1 Abbreviations

Abbreviation	Description
CAN	Controller Area Network
CANopen	(open) transmission protocol on the Controller Area Network bus
EMC	electromagnetic compatibility
ESD	Electrostatic Discharge Abbreviation for all terms relating to electrostatic discharge, e.g. ESD protection, ESD hazards, etc.
LSB	Least Significant Bit
MSB	Most Significant Bit
PDO	Project Data Object
SDO	Service Data Object

A.2 Index

A.3 Register: identification numbers in numerical order

C-48B6h	mc_positioning_velocity	15–11
	mc_positioning_velocity_1 ... mc_positioning_velocity_32	15–11
C-48C0h	mc_positioning_acceleration	15–11
	mc_positioning_acceleration_1 ... mc_positioning_acceleration_32	15–11
C-48CAh	mc_positioning_deceleration	15–12
	mc_positioning_deceleration_1 ... mc_positioning_deceleration_32	15–12
C-48D4h	mc_positioning_controls	15–12
	mc_positioning_control_1 ... mc_positioning_control_32	15–12
C-6040h	control_word	6–6
C-6041h	status_word	6–9
C-6042h	vl_target_velocity	14–3
C-6043h	vl_velocity_demand	14–4
C-6044h	vl_control_effort	14–10
C-6046h	vl_velocity_min_max_amount	14–5
	vl_velocity_min_amount	14–6
	vl_velocity_max_amount	14–6
C-6048h	vl_velocity_acceleration	14–7
	delta_speed	14–7
	delta_time	14–7
C-6049h	vl_velocity_deceleration (= C-6048h)	14–8
	delta_speed	14–8
	delta_time	14–8
C-604Ah	vl_velocity_quick_stop	14–9
	delta_speed	14–9
	delta_time	14–9
C-604Ch	vl_dimension_factor	14–4
	vl_dimension_factor_numerator	14–5
	vl_dimension_factor_denominator	14–5
C-605Ah	quick_stop_option_code	6–13
C-605Bh	shutdown_option_code	6–12
C-605Ch	disable_operation_option_code	6–12
C-605Dh	stop_option_code	6–13
C-605Eh	fault_reaction_option_code	6–14
C-6060h	modes_of_operation	6–14
C-6061h	modes_of_operation_display	6–15
C-6062h	position_demand_value	10–9
C-6063h	position_actual_value*	10–9
C-6064h	position_actual_value	10–10

C-6065h	following_error_window	10–10
C-6067h	position_window	10–11
C-606Ah	sensor_selection_code	12–4
C-606Bh	velocity_demand_value	12–5
C-606Ch	velocity_actual_value	12–5
C-606Dh	velocity_window	12–6
C-6071h	target_torque	13–9
C-6072h	max_torque	13–10
C-6076h	motor_rated_torque	13–10
C-6077h	torque_actual_value	13–11
C-607Ah	target_position	8–4
C-607Ch	home_offset	9–4
C-607Dh	software_position_limit	8–5
	min_position_limit	8–5
	max_position_limit	8–5
C-607Fh	max_profile_velocity	8–6
C-6080h	max_motor_speed	8–6
C-6081h	profile_velocity	8–7
C-6083h	profile_acceleration	8–7
C-6084h	profile_deceleration	8–8
C-6085h	quick_stop_deceleration	8–8
C-6086h	motion_profile_type	8–9
C-6087h	torque_slope	13–11
C-6088h	torque_profile_type	13–12
C-6089h	position_notation_index	7–2
C-608Ah	position_dimension_index	7–3
C-608Bh	velocity_notation_index	7–4
C-608Ch	velocity_dimension_index	7–5
C-608Dh	acceleration_notation_index	7–6
C-608Eh	acceleration_dimension_index	7–7
C-6091h	gear_ratio	7–8
	motor_revolutions	7–8
	shaft_revolutions	7–8
C-6092h	feed_constant	7–9
	feed	7–9
	shaft_revolutions	7–9
C-6098h	homing_method	9–4
C-6099h	homing_speeds	9–5
	speed_during_search_for_switch	9–5
	speed_during_search_for_zero (= C-6099.01)	9–5

Notes:

Bosch Automation Technology

Australia
Robert Bosch (Australia) Pty. Ltd.
Head Office
Cnr. Centre - McNaughton Roads
P.O. Box 66
AUS-3168 Clayton, Victoria
Fax (03) 95 41 77 03

Great Britain
Robert Bosch Limited
Automation Technology Division
Meridian South
Meridian Business Park
GB-Braunstone Leicester LE3 2WY
Fax (01 16) 289 2878

Canada
Robert Bosch Corporation
Automation Technology Division
6811 Century Avenue
CAN-Mississauga, Ontario L5N 1R1
Fax (905) 5 42-42 81

USA
Robert Bosch Corporation
Automation Technology Division
Fluid Power Products
7505 Durand Avenue
USA-Racine, Wisconsin 53406
Fax (414) 5 54-81 03

Robert Bosch Corporation
Automation Technology Division
Factory Automation Products
816 East Third Street
USA-Buchanan, MI 49107
Fax (616) 6 95-53 63

Robert Bosch Corporation
Automation Technology Division
Industrial Electronic Products
40 Darling Drive
USA-Avon, CT 0 60 01-42 17
Fax (860) 4 09-70 80

Your concessionary

BOSCH

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Fax +49 (0) 60 62 78-4 28

1070 066 095-101 (01.07) GB · HB AN · AT/PLS · Printed in Germany