Bonfiglioli
Vectron

ACTIVE CUBE

CANopen
Communication module CM-CAN
Frequency inverter 230 V / 400 V

Bonfiglioli
power, control and green solutions

Contents

1 General Information about the Documentation

For better clarity, the documentation of the frequency inverter is structured according to the customer-specific requirements.

This documentation was written in German language. The German documentation is the original one. Other language versions are translated.

Quick Start Guide

The “Quick Start Guide” describes the basic steps required for mechanical and electrical installation of the frequency inverter. The guided commissioning supports you in the selection of necessary parameters and the configuration of the software of the frequency inverter.

User manual

The user manual documents the complete functionality of the frequency inverter. The parameters required for special purposes, for adjustment to the application and the numerous additional functions are described in detail.
Separate user manuals are supplied for optional components for the frequency inverter. These manuals complement the operating instructions and the “Quick Start Guide” for the frequency inverter.

Application manual

The application manual complements the documentation to ensure goal-directed installation and commissioning of the frequency inverter. Information on various topics in connection with the use of the frequency inverter are described in context with the specific application.

Installation instructions

The installation manual describes the installation and use of devices, complementing the “Quick Start Guide” and the user manual.

1.1 This document

The present user manual of the CM-CAN communication module complements the user manual and the “Quick Start Guide” for the frequency inverters of the ACU 201 and ACU 401 device series.
The user manual contains important information on the installation and use of the CANopen® communication module CM-CAN in its specified application range. Compliance with this user manual contributes to avoiding risks, minimizing repair cost and downtimes and increasing the reliability and service live of the frequency inverter.
For this reason, make sure you read the user manual carefully.

⚠ WARNING

Compliance with the documentation is required to ensure safe operation of the frequency inverter. BONFIGLIOLI VECTRON GmbH shall not be held liable for any damage caused by any non-compliance with the documentation.

In case any problems occur which are not covered by the documentation sufficiently, please contact the manufacturer.

1.2 Warranty and liability

BONFIGLIOLI VECTRON GmbH would like to point out that the contents of this user manual do not form part of any previous or existing agreement, assurance or legal relationship. Neither are they intended to supplement or replace such agreements, assurances or legal relationships. Any obligations of the manufacturer shall solely be based on the relevant purchase agreement which also includes the complete and solely valid warranty stipulations. These contractual warranty provisions are neither extended nor limited by the specifications contained in this documentation.
The manufacturer reserves the right to correct or amend the specifications, product information and omissions in these operating instructions without notice. The manufacturer shall not be liable for any damage, injuries or costs which may be caused by the aforementioned reasons.

In addition to that, BONFIGLIOLI VECTRON GmbH excludes any warranty/liability claims for any personal and/or material damage if such damage is due to one or more of the following causes:

- inappropriate use of the frequency inverter,
- non-compliance with the instructions, warnings and prohibitions contained in the documentation,
- unauthorized modifications of the solar inverter,
- insufficient monitoring of parts of the machine/plant which are subject to wear,
- repair work at the machine/plant not carried out properly or in time,
- catastrophes by external impact and Force Majeure.

1.3 Obligation

This user manual must be read before commissioning and complied with. Anybody entrusted with tasks in connection with the

- transport,
- assembly,
- installation of the frequency inverter and
- operation of the frequency inverter

must have read and understood the user manual and, in particular, the safety instructions in order to prevent personal and material losses.

1.4 Copyright

1.5 Storage

The documentation form an integral part of the frequency inverter. It must be stored such that it is accessible to operating staff at all times. In case the frequency inverter is sold to other users, this user manual must also be handed over.

2 General safety instructions and information on use

The chapter "General safety instructions and information on use" contains general safety instructions for the Operator and the Operating Staff. At the beginning of certain main chapters, some safety instructions are included which apply to all work described in the relevant chapter. Special work-specific safety instructions are provided before each safety-relevant work step.

2.1 Terminology

According to the documentation, different activities must be performed by certain persons with certain qualifications.
The groups of persons with the required qualification are defined as follows:

Operator
This is the entrepreneur/company who/which operates the frequency inverter and uses it as per the specifications or has it operated by qualified and instructed staff.

Operating staff
The term Operating Staff covers persons instructed by the Operator of the frequency inverter and assigned the task of operating the frequency inverter.

Qualified staff
The term Qualified Staff covers staff who is assigned special tasks by the Operator of the frequency inverter, e.g. installation, maintenance and service/repair and troubleshooting. Based on their qualification and/or know-how, qualified staff must be capable of identifying defects and assessing functions.

Qualified electrician
The term Qualified Electrician covers qualified and trained staff who has special technical know-how and experience with electrical installations. In addition, Qualified Electricians must be familiar with the applicable standards and regulations, they must be able to assess the assigned tasks properly and identify and eliminate potential hazards.

Instructed person
The term Instructed Person covers staff who was instructed and trained about/in the assigned tasks and the potential hazards that might result from inappropriate behavior. In addition, instructed persons must have been instructed in the required protection provisions, protective measures, the applicable directives, accident prevention regulations as well as the operating conditions and verified their qualification.

Expert
The term Expert covers qualified and trained staff who has special technical know-how and experience relating to frequency inverter. Experts must be familiar with the applicable government work safety directives, accident prevention regulations, guidelines and generally accepted rules of technology in order to assess the operationally safe condition of the frequency inverter.

2.2 Designated use

The frequency inverter is designed according to the state of the art and recognized safety regulations.

The frequency inverters are electrical drive components intended for installation in industrial plants or machines. Commissioning and start of operation is not allowed until it has been verified that the machine meets the requirements of the EC Machinery Directive 2006/42/EC and DIN EN 60204-1.

The frequency inverters meet the requirements of the low voltage directive 2006/95/EEC and DIN EN 61800-5-1. CE-labeling is based on these standards. Responsibility for compliance with the EMC Directive 2004/108/EC lies with the operator. Frequency inverters are only available at specialized dealers and are exclusively intended for commercial use as per EN 61000-3-2.

No capacitive loads may be connected to the frequency inverter.

The technical data, connection specifications and information on ambient conditions are indicated on the rating plate and in the documentation and must be complied with in any case.

2.3 Misuse

Any use other than that described in "Designated use" shall not be permissible and shall be considered as misuse.
For example, the machine/plant must not be operated

- by uninstructed staff,
- while it is not in perfect condition,
- without protection enclosure (e.g. covers),
- without safety equipment or with safety equipment deactivated.

The manufacturer shall not be held liable for any damage resulting from such misuse. The sole risk shall be borne by the operator.

2.3.1 Explosion protection

The frequency inverter is an IP 20 protection class device. For this reason, use of the device in explosive atmospheres is not permitted.

2.4 Residual risks

Residual risks are special hazards involved in handling of the frequency inverter which cannot be eliminated despite the safety-compliant design of the device. Residual risks are not obviously identifiable and can be a potential source of injury or health hazard.
Typical residual hazards include:

Electrical hazard

Danger of contact with energized components due to a defect, opened covers or enclosures or improper working on electrical equipment.
Danger of contact with energized components inside of the frequency inverter if no external disconnection device was installed by the operator.

Electrostatic charging

Touching electronic components bears the risk of electrostatic discharges.

Thermal hazards

Risk of accidents by hot machine/plant surfaces, e.g. heat sink, transformer, fuse or sine filter.

Charged capacitors in DC link

The DC link may have dangerous voltage levels even up to three minutes after shutdown.

Danger of equipment falling down/over, e.g. during transport

Center of gravity is not the middle of the electric cabinet modules.

2.5 Safety and warning signs at frequency inverter

- Comply with all safety instructions and danger information provided on the frequency inverter.
- Safety information and warnings on the frequency inverter must not be removed.

2.6 Warning information and symbols used in the user manual

2.6.1 Hazard classes

The following hazard identifications and symbols are used to mark particularly important information:

⚠ DANGER

Identification of immediate threat holding a **high** risk of death or serious injury if not avoided.

⚠ WARNING

Identification of immediate threat holding a **medium** risk of death or serious injury if not avoided.

⚠ CAUTION

Identification of immediate threat holding a **low** risk of minor or moderate physical injury if not avoided.

NOTE

Identification of a threat holding a risk of material damage if not avoided.

2.6.2 Hazard symbols

Symbol	Meaning	Symbol	Meaning
	General hazard		Suspended load
	Electrical voltage		Hot surfaces

2.6.3 Prohibition signs

Symbol	Meaning
	No switching; it is forbidden to switch the machine/plant, assembly on

2.6.4 Personal safety equipment

Symbol	Meaning
	Wear body protection

2.6.5 Recycling

Symbol	Meaning
	Recycling, to avoid waste, collect all materials for reuse

2.6.6 Grounding symbol

Symbol	Meaning
	Ground connection

2.6.7 ESD symbol

Symbol	Meaning
	ESD: Electrostatic Discharge (can damage components and assemblies)

2.6.8 Information signs

Symbol	Meaning
	Tips and information making using the frequency inverter easier.

2.7 Directives and guidelines to be adhered to by the operator

The operator must follow the following directives and regulations:

- Ensure that the applicable workplace-related accident prevention regulations as well as other applicable national regulation are accessible to the staff.
- An authorized person must ensure, before using the frequency inverter, that the device is used in compliance with its designated use and that all safety requirements are met.
- Additionally, comply with the applicable laws, regulations and directives of the country in which the frequency inverter is used.
- Any additional guidelines and directives that may be required additionally shall be defined by the operator of the machine/plant considering the operating environment.

2.8 Operator's general plant documentation

- In addition to the user manual, the operator should issue separate internal operating instructions for the frequency inverter. The user manual of the frequency inverter must be included in the user manual of the whole plant.

2.9 Operator's/operating staff's responsibilities

2.9.1 Selection and qualification of staff

- Any work on the frequency inverter may only be carried out by qualified technical staff. The staff must not be under the influence of any drugs. Note the minimum age required by law. Define the staff's responsibility in connection with all work on the frequency inverter clearly.
- Work on the electrical components may only be performed by a qualified electrician according to the applicable rules of electrical engineering.
- The operating staff must be trained for the relevant work to be performed.

2.9.2 General work safety

- In addition to the user manual of the machine/plant, any applicable legal or other regulations relating to accident prevention and environmental protection must be complied with. The staff must be instructed accordingly.
  Such regulations and/or requirements may include, for example, handling of hazardous media and materials or provision/use of personal protective equipment.
- In addition to this user manual, issue any additional directives that may be required to meet specific operating requirements, including supervision and reporting requirements, e.g. directives relating to work organization, workflow and employed staff.
- Unless approved of expressly by the manufacturer, do not modify the frequency inverter in any way, including addition of attachments or retrofits.
- Only use the frequency inverter if the rated connection and setup values specified by the manufacturer are met.
- Provide appropriate tools as may be required for performing all work on the frequency inverter properly.

2.10 Organizational measures

2.10.1 General

- Train your staff in the handling and use of the frequency inverter and the machine/plant as well as the risks involved.
- Use of any individual parts or components of the frequency inverter in other parts of the operator's machine/plant is prohibited.
- Optional components for the frequency inverter must be used in accordance with their designated use and in compliance with the relevant documentation.

2.10.2 Use in combination with third-party products

- Please note that BONFIGLIOLI VECTRON GmbH will not accept any responsibility for compatibility with third-party products (e.g. motors, cables or filters).
- In order to enable optimum system compatibility, BONFIGLIOLI VECTRON GmbH office components facilitating commissioning and providing optimum synchronization of the machine/plant parts in operation.
- If you use the frequency inverter in combination with third-party products, you do this at your own risk.

2.10.3 Transport and Storage

- The frequency inverters must be transported and stored in an appropriate way. During transport and storage the devices must remain in their original packaging.
- The units may only be stored in dry rooms which are protected against dust and moisture and are exposed to little temperature deviations only. The requirements of DIN EN 60721-3-1 for storage, DIN EN 60721-3-2 for transport and labeling on the packaging must be met.
- The duration of storage without connection to the permissible nominal voltage may not exceed one year.

2.10.4 Handling and installation

- Do not commission any damaged or destroyed components.
- Prevent any mechanical overloading of the frequency inverter. Do not bend any components and never change the isolation distances.
- Do not touch any electronic construction elements and contacts. The frequency inverter is equipped with components which are sensitive to electrostatic energy and can be damaged if handled improperly. Any use of damaged or destroyed components will endanger the machine/plant safety and shall be considered as a non-compliance with the applicable standards.
- Only install the frequency inverter in a suitable operating environment. The frequency inverter is exclusively designed for installation in industrial environments.
- If seals are removed from the case, this can result in the warranty becoming null and void.

2.10.5 Electrical connections

- The five safety rules must be complied with.
- Never touch live terminals. The DC link may have dangerous voltage levels even up to three minutes after shutdown.
- When performing any work on/with the frequency inverter, always comply with the applicable national and international regulations/laws on work on electrical equipment/plants of the country when the frequency inverter is used.
- The cables connected to the frequency inverters may not be subjected to high-voltage insulation tests unless appropriate circuitry measures are taken before.
- Only connect the frequency inverter to suitable supply mains.

2.10.5.1 The five safety rules

When working on/in electrical plants, always follow the five safety rules:

1. Isolate
2. Secure to prevent restarting
3. Check isolation
4. Earth and short-circuit,
5. Cover or shield neighboring live parts.

2.10.6 Safe operation

- During operation of the frequency inverter, always comply with the applicable national and international regulations/laws on work on electrical equipment/plants.
- Before commissioning and the start of the operation, make sure to fix all covers and check the terminals. Check the additional monitoring and protective devices according to the applicable national and international safety directives.
- During operation, never open the machine/plant
- Do not connect/disconnect any components/equipment during operation.
- The machine/plant holds high voltage levels during operation, is equipped with rotating parts (fan) and has hot surfaces. Any unauthorized removal of covers, improper use, wrong installation or operation may result in serious injuries or material damage.
- Some components, e.g. the heat sink or brake resistor, may be hot even some time after the machine/plant was shut down. Don't touch any surfaces directly after shutdown. Wear safety gloves where necessary.
- The frequency inverter may hold dangerous voltage levels until the capacitor in the DC link is discharged. Wait for at least 3 minutes after shutdown before starting electrical or mechanical work on the frequency inverter. Even after this waiting time, make sure that the equipment is deenergized in accordance with the safety rules before starting the work.
- In order to avoid accidents or damage, only qualified staff and electricians may carry out the work such as installation, commissioning or setup.
- In the case of a defect of terminals and/or cables, immediately disconnect the frequency inverter from mains supply.
- Persons not familiar with the operation of frequency inverters must not have access to the frequency inverter. Do not bypass nor decommission any protective facilities.
- The frequency inverter may be connected to power supply every 60 s. This must be considered when operating a mains contactor in jog operation mode. For commissioning or after an emergency stop, a non-recurrent, direct restart is permissible.
- After a failure and restoration of the power supply, the motor may start unexpectedly if the Auto Start function is activated.
  If staff is endangered, a restart of the motor must be prevented by means of external circuitry.
- Before commissioning and the start of the operation, make sure to fix all covers and check the terminals. Check the additional monitoring and protective devices according to EN 60204 and applicable the safety directives (e.g. Working Machines Act or Accident Prevention Directives).

2.10.7 Maintenance and service / troubleshooting

- Visually inspect the frequency inverter when carrying out the required maintenance work and inspections at the machine/plant.
- Perform the maintenance work and inspections prescribed for the machine carefully, including the specifications on parts/equipment replacement.
- Work on the electrical components may only be performed by a qualified electrician according to the applicable rules of electrical engineering. Only use original spare parts.
- Unauthorized opening and improper interventions in the machine/plant can lead to personal injury or material damage. Repairs on the frequency inverters may only be carried out by the manufacturer or persons authorized by the manufacturer. Check protective equipment regularly.
- Before performing any maintenance work, the machine/plant must be disconnected from mains supply and secured against restarting. The five safety rules must be complied with.

2.10.8 Final decommissioning

Unless separate return or disposal agreements were made, recycle the disassembled frequency inverter components:

- Scrap metal materials
- Recycle plastic elements
- Sort and dispose of other component materials

Electric scrap, electronic components, lubricants and other utility materials must be treated as special waste and may only be disposed of by specialized companies.

In any case, comply with any applicable national disposal regulations as regards environmentally compatible disposal of the frequency inverter. For more details, contact the competent local authorities.

3 Introduction

This document describes the features of the CANopen® communication for frequency inverters of the ACU series.
The CANopen® communication (like described in this manual) requires software version 5.1.2 or higher.
“Profile velocity” mode is supported with software version 5.2.0 or higher.
“Move away from Limit Switch” mode and “Electronic Gear: Slave” mode is supported with software version 5.3.0 or higher.
“Cyclic Synchronous Positioning” mode and “Cyclic Synchronous Positioning” mode is Supported with software version 5.4.0 or higher.

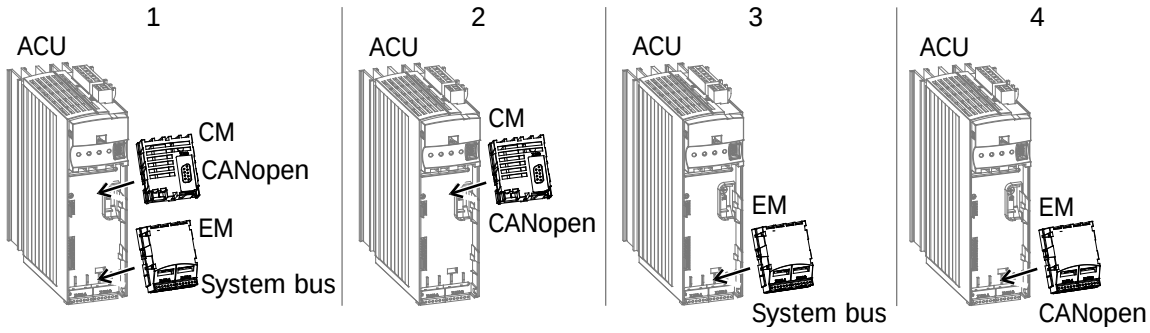
CANopen® communication is available with modules:

- Communication module CM-CAN
- Expansion module EM-SYS
- Expansion module EM with CAN terminals on board like EM-IO-01

The expansion modules can be used with either System bus or CANopen®, depending of the selection of the CANopen® interface.

Please refer to chapter 16.6 “Object support in the Software versions and EDS files” for detailed information about the supported objects and required EDS files.

Possible combinations



The frequency inverter must be extended by either the CANopen® communication module CM-CAN or a fitting EM module for the CAN connection.
The CM-CAN CANopen® module is enclosed with the frequency inverter as a separate component and must be fitted by the user. This is described in detail in the "Assembly" chapter. For the assembly of the EM modules and System bus protocol description refer to the corresponding manual.

NOTE

CM-CAN offers decoupled drivers, while EM modules have coupled drivers. BONFIGLIOLI VECTRON recommends using the CM-CAN module, especially in environments with critical EMC behaviour.

For reasons of better readability, in the following chapters CM-CAN module is used representative for all modules able to establish CANopen® communication.

These instructions are not to be understood as fundamental information on CANopen®. They presuppose underlying knowledge of the methods and mode of effect of CAN open on the part of the user.

In some chapters, as an alternative to the KP500 control unit, the setting and displaying of values is described with the help of the VPlus control software. Operation of a PC with the VPlus control software requires an optional KP232 interface adapter.

In this document, connecting the hardware, relevant parameters and the available objects are shown.

The available objects are sub-divided according to:

Communication objects	(0x1nnn)	to DS301 V4.01
Manufacturer objects	(0x2nnn)	
Standardized objects	(0x6nnn)	to DS402 V1.1

The functions and objects are described as far as necessary in these instructions. For further information, reference is made here to the Draft Standards of the CiA®.
The standards to which reference is made are DS102, DS301 and DS402, which are available from:

CiA, CAN in AUTOMATION
Am Weichselgarten 26 **Tel.: +49 9131 69086-0**
D-91058 Erlangen **Fax: +49 9131 69086-79**

NOTE

With the CM-EtherCAT communication module, it is possible to access **ALL** frequency inverters parameters from a controller. There is no access control via the control level as in the case of the KP500 manual control unit or the VPlus PC software. Changing parameters, the functions of which are not known to the user, can result in unintended movements and material and/or personal losses as well as inoperativeness of the frequency inverter.

NOTE

If data is written cyclically comply with the instructions in chapter 12.3.1 “Handling of data sets/cyclic writing”.

For the operation with a PLC in most cases an EDS file in required. You can find this EDS file on the product documentation CD.

CANopen® and CiA® are registered community trademarks of CAN in Automation e.V..

Hexadecimal values are marked in the following by a preceding “0x”.

3.1 Supported Configurations

ACTIVE CUBE inverters support different types of control and reference values:

- Standard (without Positioning functions)
- Positioning via contacts (or remote contacts)
- Positioning via Motion Control Interface (MCI) via field bus

Motion control configurations are set when parameter *configuration* **30** = x40 (in example 240). To use the full functionality of the Motion Control Interface Parameter *Local/Remote* **412** = "1-Control via State machine" must be set.

The inverter's behaviour with respect to *control word* / *status word* and *modes of operation* / *modes of operation display* is different in the two different types of configuration.

Standard:
Necessary settings: *Configuration* **30** ≠ x40.
Local/Remote **412** = (Remote) contacts

➔ The control (Start, Stop, Frequency change over, etc.) is carried out typically via:
- Digital contacts
- Remote contacts via Field bus

➔ Reference values result from the select configuration. Typical are:
- Reference speed / Reference frequency:
  - Analogue input
  - Fixed values from parameters
  - 0x6042 target velocity
- Percentage reference value for technology controller or Torque control
  - Analogue input
  - Fixed values from parameters

Please refer to chapter 14.3 "Non motion control configurations"for the control without Positioning functionality.

Positioning via contacts (or remote contacts):
Necessary settings: *Configuration* **30** = x40.
Local/Remote **412** = (Remote) contacts

➔ The control (Start, Stop, Target position change over, etc.) is carried out typically via:
- Digital contacts
- Remote contacts via Field bus

➔ Reference values result from the selected configuration. Typical are:
- Reference speed / Reference frequency
- Reference target position

Please refer also to the application manual "Positioning".

MCI (Motion Control Interface – Positioning via Field bus):
Necessary settings: *Configuration* **30** = x40.
Local/Remote **412** = 1 - Statemachine

➔ The control (Start, Stop, mode change over, etc.) is carried out via 0x6040 *Control word*.
➔ Reference values result from the selected 0x6060 *Modes of Operation*.
Typical are:
- Reference speed via 0x6042 target velocity
- Target position 0x607A target position.

The usage of the Motion Control Interface is described in this manual in chapter 14.4 "Motion control configurations".

4 First Commissioning

For the first commissioning you should acquaint yourself with the following steps and the described functions:

- Installation of the Module — Chapter 5.1
- Check and set Terminating resistor — Chapter 6
- Setting of the Baudrate — Chapter 7
- Setting of the node Address — Chapter 8
- Select the device control *Local/Remote* **412** — Chapter 14
- Commission the device function via PLC
  - Reaction of the PLC to the Boot-Up message or Guarding-Request/Response — Chapter 11.7.2, 11.8
  - PDO Mapping — Chapter 12.2.19, 12.2.20, 12.2.21, 12.2.22
  - Fault reaction — Chapter 10, 12.2.17, 12.5.1
    - Fault reset — Chapter 11.11
  - SDO access — Chapter 11.3, 12.3
- Setting Reference value:
  - Speed setting in speed controlled configuration x10, x11, x15, x16, x30, x60 — Chapter 14.3
  - Reference value in Positioning configuration x40 — Chapter 13 and 14.4
    - Velocity Mode — Chapter 14.4.1
    - Profile Velocity Mode — Chapter 14.4.2
    - Profile Position Mode — Chapter 14.4.3
    - Homing Mode — Chapter 14.4.4
    - Interpolated Position Mode — Chapter 14.4.5
    - Cyclic Synchronous Position Mode — Chapter 14.4.6
    - Cyclic Synchronous Velocity Mode — Chapter 14.4.7
    - Table Travel record Mode — Chapter 14.4.8
    - Move Away from Limit Switch — Chapter 14.4.9
    - Mode of Operation change — Chapter 12.5.12
- Diagnosis: — Chapter 16, 15

5 Installation/Disassembly of the communication module

5.1 Installation

The communication module CM-CAN is pre-assembled in a case. Additionally, a PE spring is enclosed for PE connection (shield).

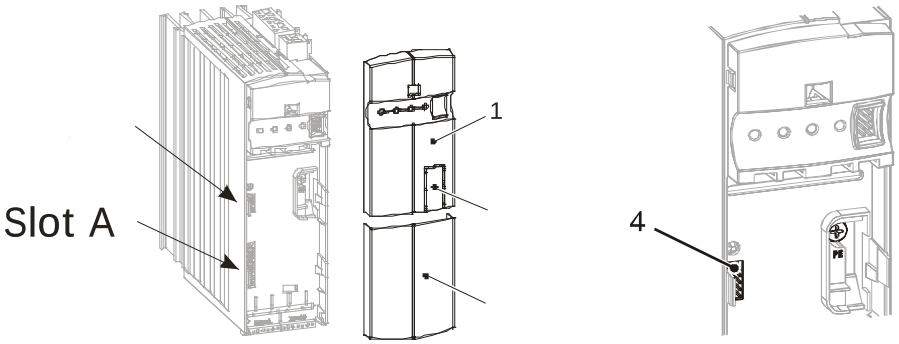
⚠ Caution!

Danger of destroying the frequency inverter and/or the communication module

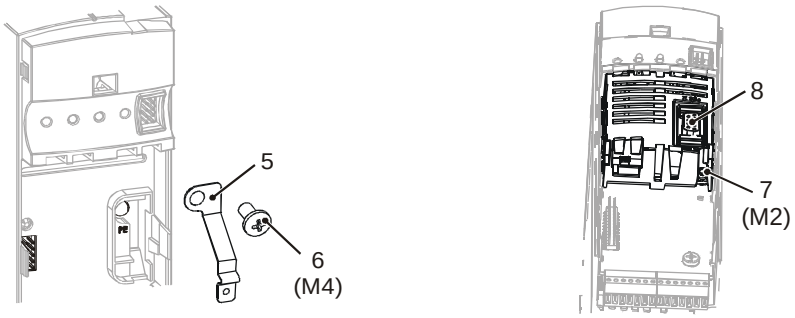
- The frequency inverter must be disconnected from the power supply before installation of the communication module. Assembly under voltage is not permissible.
- Do not touch the PCB visible on the back of the module, otherwise components may be damaged.

Work steps:

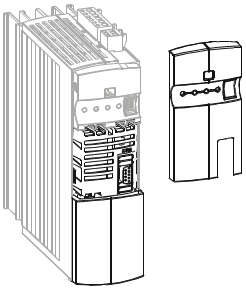
- Disconnect the frequency inverter from the mains voltage and protect it against being energized unintentionally.
- Remove covers **(1)** and **(2)** of the frequency inverter. Slot B **(4)** for the communication module is now accessible.



- Mount the supplied PE spring **(5)** using the M4 screw **(6)** in the unit. The spring must be aligned centrally.
- Insert the communication module in slot B **(4)** until it engages audibly.
- Fix the communication module by screwing the M2 screw **(7)** of the module to the PE spring **(5)**.

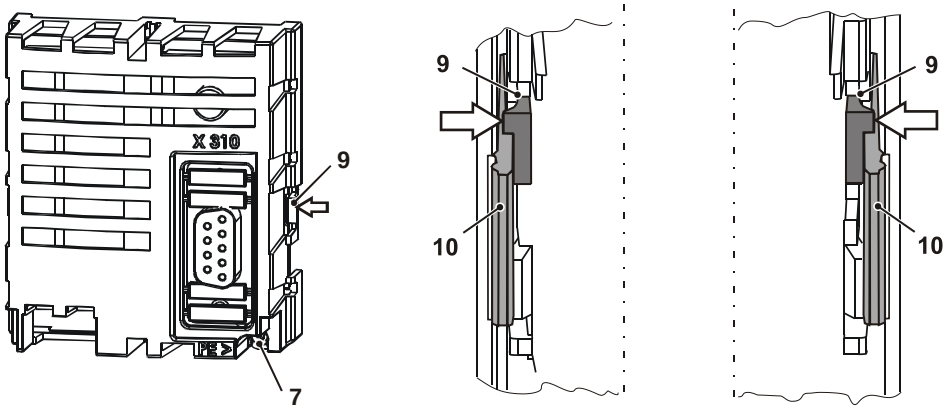


- In the upper cover **(1)**, break out the pre-punched cutout **(3)** for the plug X310 **(8)**.
- Mount the two covers **(1)** and **(2)**.



5.2 Disassembly

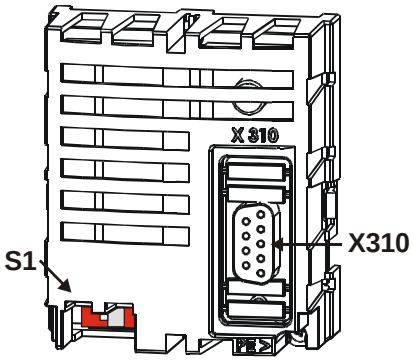
- Disconnect the frequency inverter from mains voltage and protect it against being energized unintentionally.
- Remove covers **(1)** and **(2)** of the frequency inverter.



- Loosen the M2 screw **(7)** on the communication module
- Unplug the communication module from Slot B **(4)** by unlocking the locking hooks **(9)** on the right and left hand side of the module from the case of the frequency inverter using a small screwdriver.
  The looking hooks **(9)** are located at the place where the looking hooks **(10)** for the upper cover **(1)** project from the case of the frequency inverter.
  - To do this, carefully insert the screwdriver in the gap between the case of the module and the frequency inverter and push the locking hook inwards in the direction of the arrow **(⇦)**. As soon as the right hand side is unlocked, pull the module out a bit on the right hand side and hold it.
  - Hold the module on the right hand side while unlocking the locking hook on the left hand side in the same way **(⇨)**.
  - Pull the module out of the slot by gently pulling on the right and left hand side alternately.
- Disassemble the PE spring **(5)**.
- Mount the two covers **(1)** and **(2)**.

6 Connector pin assignment / bus termination / line

The CAN connection is physically designed according to the ISO 11898 standards (CAN High-Speed).

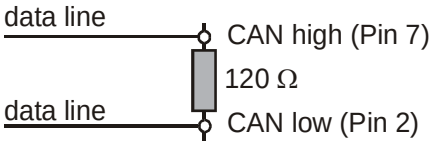


The **X310** (9-pole Sub-D) bus plug has been designed according to DS102 Version 2.0 (Bus node, option A).
Details can be seen from the following table on the occupancy of the bus plug.

The bus termination necessary on a phase in the physically first and last subscriber can be activated via **DIP switch S1** on the communication module.

The factory setting for the bus termination is OFF.

As an alternative, this is also possible via corresponding switching in the bus connection plugs.



NOTE

Make absolutely sure that only one of the two possibilities for the bus termination is used and the bus termination is only switched on with the first and last subscriber. Otherwise, operation of the CANopen® communication is not possible. The CAN Controller State is displayed via actual value parameter *CAN-State* **1291**.

Bus plug X310		
Pin	**Name**	**Function**
Housing	Shield	connected with PE
1	CAN_L	CAN Low bus interface, short-circuit resistant and function-insulated, max. current 60 mA
2	CAN_L	CAN Low bus interface, short-circuit resistant and function-insulated, max. current 60 mA
3	CAN_GND	Earth/GND
4	n.c.	not used
5	n.c.	not used
6	CAN_GND	Earth/GND
7	CAN_H	CAN High bus interface, short-circuit resistant and function-insulated, max. current 60 mA
8	CAN_H	CAN-High Bus-interface, short-circuit resistant and function-insulated, max. current 60 mA
9	-	Do NOT connect.

The drilled and shielded line is to be used for the bus line. The shield is to be implemented as a harness shield (**not a film shield**).

NOTE

Connect the line screen with PE at both ends.

7 Baud rate setting / line lengths

The transmission speed of the CANopen® communication module CM-CAN can be set via the parameter *CAN Baud rate* **385**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
385	CAN Baud rate	1	8	6

The transmission rate is a function of a variety of application-specific parameters. The line length of the communication network limits the transmission speed due to the signal propagation time of the CANopen® protocols.

CANopen® interface		
Operation mode	**Function**	**max. Line length**
1 - 10 kBaud	Transmission rate 10 kBaud	5000 meter
2 - 20 kBaud	Transmission rate 20 kBaud	2500 meter
3 - 50 kBaud	Transmission rate 50 kBaud	1000 meter
4 - 100 kBaud	Transmission rate 100 kBaud	500 meter
5 - 125 kBaud	Transmission rate 125 kBaud	500 meter
6 - 250 kBaud	Transmission rate 250 kBaud	250 meter
7 - 500 kBaud	Transmission rate 500 kBaud	100 meter
8 - 1000 kBaud	Transmission rate 1000 kBaud	25 meter

Changing the baud rate causes a restart of the CANopen® system (NOT a reset of the inverter).

8 Setting the node number

The CANopen® protocol supports a maximum of 127 nodes in a communication network. Each frequency inverter is assigned a node ID, which may only exist once in the system, for its unambiguous identification. The node number is set with parameter *CAN Node Number* **387**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
387	CAN Node Number	-1	127	-1

The factory setting *CAN Node Number* **387** = -1 means that the CANopen® interface has been **deactivated**.

The value *Can Node number* **387** = 0 is not allowed and cannot be set.

Changing the node number causes a restart of the CANopen® system (NOT a reset of the inverter).

9 Assigning the CANopen interface

Normally a CANopen® connection is set up using the CM-CAN module. As an alternative for special applications, the CANopen® connection can be switched to an EM-module with a CAN driver connection via parameter *CAN-Interface* **276**. The interface can only be changed when parameter *Node-Id* **900** of the system bus is set to value **-1** and *CAN Node Number* **387** is set to **-1**.

CAN Interface **276**	
Operation mode	**Function**
1 - CM-CAN	CM-CAN is used for the CANopen® connection. Factory setting.
2 - EM-xxx	EM-xxx is used for the CANopen® connection

The setting of *CAN interface* **276** = 2 is only possible when an EM-module with CAN system bus is installed.

Even if only an EM-module with CAN system bus is installed, **276** first displays the value "1 – CM-CAN" which must then be changed to "2 – EM-xxx" in order to activate the EM-module for the CANopen® connection.

If *CAN interface* **276** = 2 is set, the transmission speed is set with *CAN Baud rate* **385**. *Baud Rate* **903** (System bus) is deactivated by setting **900** = -1. The same applies to all other parameters which have a function when using the system bus.
If *CAN interface* **276** = 2 is set, a parallel Systembus operation is not possible. In this case the parameter *Node-ID* **900** cannot be changed deviating from “-1”.

10 Operational behaviour on bus failure

The operational behaviour if the CANopen® system fails due to BusOff, guarding, heartbeat, SYNC error, RxPDO length error or NMT state change (leaving NMT state operational) can be parameterized. The required behaviour is set with parameter *Bus Error Behaviour* **388**. or via Object 0x6007 abort connection option code.

For the description of the inverter's functional behaviour, see chapter 12.5.1 “0x6007/0 Abort Connection option code”.

CAN Error Behaviour **388**	Function
0 - No Reaction	Operating point is maintained
1 - Error	Device state machine changes immediately to state “fault” (**factory setting**)
2 - Switch-off	Device state machine processes command ‘*disable voltage*’ and changes to state “switch on disabled”
3 - Quick-Stop	Device state machine processes command ‘*quick stop*’ and changes to state “switch on disabled”
4 - Ramp-Stop + Error	Device state machine processes command ‘*disable operation*’ and changes to state “fault” after the drive is stopped
5 - Quick-Stop + Error	Device state machine processes command ‘*quick stop*’ and changes to state “fault” after the drive is stopped

NOTE

The parameter settings *CAN Error Behaviour* **388** = 2 ... 5 are evaluated depending of parameter *Local/Remote* **412**. This is described in detail in chapter 12.5.1 “0x6007/0 Abort Connection option code”.

Parameter *CAN Error Behaviour* **388** corresponds to the device profile object 0x6007 abort connection option code.

The error and warning behaviour of the frequency inverter can be parameterized in various ways. Occurring errors are described in detail in chapter 16.4 “Fault messages”.

NOTE

The disconnection of a connector or another contact loss can only be detected safely via set up timeout monitoring.

11 CANopen Overview

CANopen® is used in a wide range of applications and is an especially favoured communication system for motion control applications. The CANopen® based standard DS402 “drives and motion control” describes and defines the necessary objects and functions for motion control systems.

The CANopen® standard DS301 describes the basic communication functions in principle. This chapter will give a short overview of the different functions based on DS301. Detailed information on the CAN physical layer and CANopen® DS301 functions can be found in the respective literature (e. g. “Controller Area Network” by Prof. Dr.-Ing. K. Etschberger) and standards published by CAN-in-Automation CiA® (www.can-cia.org).

Every CANopen® device contains an object dictionary with all supported objects. The objects can be divided into the two main groups – communication objects and application objects. The objects are addressed by their index 0xnnnn (16 bit) and subindex 0xnn (8 bit).

The different functions defined by CANopen® (NMT, SDO, SYNC, PDO, Emergency) use fixed identifier ranges. These identifier ranges are defined by the “Predefined Connection Set”. By default every function uses an identifier calculated as the base number plus node-ID (node–ID set by parameter *CAN node number* **387**.

11.1 Communication Objects

The communication objects are located in the index range 0x1nnn. They describe the communication behaviour of a CANopen® device. Some of the communication objects comprise device information
(e. g. manufacturer’s vendor-id or inverter serial number). With the help of communication objects the application objects for device control are mapped to the PDO messages.

11.2 Application Objects

The application objects are divided into two groups again. The index range 0x2000 – 0x5FFF is reserved for manufacturer specific objects and the index range 0x6nnn is reserved for device profile specific objects. Device profile specific objects 0x6nnn are defined by DS402 drives and motion control. They are used for controlling the device application (start/stop, speed, motion control functions).

11.3 SDO Function

The SDO (Service Data Objects) messages are used for reading and writing the objects located in the object dictionary. Objects with up to four bytes of data are transferred with an expedited SDO transfer that uses one request and one response message. Access to objects with more than four bytes of data is accomplished by a segmented domain transfer.

In chapter 12.3 “Manufacturer objects (0x2nnn)” the necessary messages for reading/writing objects with expedited transfer are described in detail. Access to communication, manufacturer and device profile specific objects with up to four bytes of data is accomplished in the same way. The only difference is in the index and sub-index number.

The inverter supports one server SDO. This server SDO is accessed by the client SDO on the PLC side. An SDO message always has a COB-ID followed by 8 data bytes.

SDO-message:

COB-ID	0	1	2	3	4	5	6	7
COB-ID	command specifier (cs)	index		sub-index	data	data	data	data
	nn	LSB	MSB					

Default Identifiers (COB-ID):

TxSDO 0x600 (=1536) + Node-ID
RxSDO 0x580 (=1408) + Node-ID

Depending on the transfer direction and the amount of data bytes, different command specifiers are used.
The error codes of failed SDO accesses are listed in chapter 11.3.3.

11.3.1 Read Access

Client → Server, Upload Request

COB-ID	0	1	2	3	4	5	6	7
0x600 + Node-ID	cs	index		sub- index	data	data	data	data
	0x40	LSB	MSB		00	00	00	00

Server → Client, Upload Response

COB-ID	0	1	2	3	4	5	6	7
0x580 + Node-ID	cs	index		sub- index	data	data	data	data
	0x4x	LSB	MSB		data01	data02	data03	data04

The amount of valid data bytes is coded in the response of the command specifier.

Amount of data bytes	**1**	**2**	**3**	**4**
Command specifier (cs)	0x4F	0x4B	0x47	0x43

Examples for SDO parameter read access are described in chapter 12.3.1.3 “Examples of reading parameters”.

11.3.2 Write Access

Client → Server, Download Request

COB-ID	0	1	2	3	4	5	6	7
0x600 + Node-ID	cs	index		sub- index	data	data	data	data
	0x2x	LSB	MSB		data01	data02	data03	data04

Server → Client, Download Response

COB-ID	0	1	2	3	4	5	6	7
0x580 + Node-ID	cs	index		sub- index	data	data	data	data
	0x60	LSB	MSB		00	00	00	00

The amount of valid data bytes must be coded in the request of the command specifier.

Amount of data bytes	**1**	**2**	**3**	**4**
Command specifier	0x2F	0x2B	0x27	0x23

NOTE

Using Write accesses for parameters (objects 0x2nnn = index), the sub-index is used to define the Write access into EEPROM or RAM. Please refer to chapter 12.3.1 “Handling of data sets/cyclic writing”.

Examples for SDO parameter write access are described in chapter 12.3.1.2 “Examples of writing parameters”.

11.3.3 Error code table

If an error occurs in reading or writing, the server SDO of the frequency inverter replies with the SDO abort message. This message contains the index/subindex and appropriate error code.

Server ➔ Client Abort SDO Transfer

COB-ID	0	1	2	3	4	5	6	7
0x580 + Node-ID	cs	index		sub- index	abort code low		abort code high	
	0x80	LSB	MSB	LSB	MSB	LSB	MSB	00

Error codes			
Abort code high	**Abort code low**	**Description to CANopen®**	**Product-specific allocation**
0x0504	0x0000	SDO protocol timed out	SDO access Time Out
0x0601	0x0000	Unsupported access to an object	- Parameter cannot be written or read
0x0602	0x0000	Object does not exist	- Parameter does not exist
0x0604	0x0047	General internal incompatibility in the device	- Data sets differ
0x0606	0x0000	Access failed due to a harware error	- EEPROM Error (Read/write/checksum)
0x0607	0x0010	Data type does not match	- Parameter has a different data type
0x0607	0x0012	Data type does not match or length of Service telegram too big	Parameter has a different data type or telegram length not correct.
0x0607	0x0013	Data type does not match or length of Service telegram too small	Parameter has a different data type or telegram length not correct.
0x0609	0x0011	Subindex does not exist	- Data set does not exist
0x0609	0x0030	Value range of parameter exceeded	- Parameter value too large or too small
0x0609	0x0031	Value of parameter written too high.	- Parameter value too large
0x0609	0x0032	Value of parameter written too low.	- Parameter value too small
0x0800	0x0020	Data cannot be transmitted or saved	- Invalid value for operation
0x0800	0x0021	Data cannot be transferred because of local control	- parameter cannot be written in operation

11.3.4 Segmented Transfer

For data lengths > 4 Bytes the so called Segmented Transfer is used – the expedited Transfer only supports lengths up to 4 Bytes.

In the first "Initiate" Telegram the overall amount of used data of the following sequence telegrams is defined.
In the following telegrams 7 data byes per telegram are send until the amount of data bytes to be transmitted was reached. The sequences of the Segmented Transfer are separated by a toggle bit in the command specifier for the request and the reply telegram. A "Continue" Bit marks the last telegram.

11.3.4.1 Reading „Segmented Transfer"

When Reading a regular Read access via command specifier 0x40 is executed. The response contains the command specifier 0x41 that marks the requirement of Segmented Transfer for this object. The following requests alternate with command specifiers 0x60 and 0x70 until all data bytes were transmitted. In the last segment the command specifier (bits 1...3) contains the amount of not used data bytes in that last segment.
The resulting request and response telegrams are shown in the following sequence.

The Command Specifier have the following setup:

Initiate Upload Command Specifier:

Request:	Bit	7	6	5	4	3	2	1	0
		ccs			0	0	0	0	0
Response:	Bit	7	6	5	4	3	2	1	0
		scs			0	n		e	s

Segment Upload Command Specifier:

Request:	Bit	7	6	5	4	3	2	1	0
		ccs			t	0	0	0	0
Response:	Bit	7	6	5	4	3	2	1	0
		scs			t	n			c

Abbreviation	Description	Values
ccs	Client command Specifier	2 = Initiate upload request 3 = Upload segment request
scs	Server command Specifier	2 = Initiate upload response 0 = Upload segment response
n	Only valid if e =1 AND s = 1, in all other cases n = 0.	If valid: Amount of data bytes, that contain no useful data
e	Transfer type	0 = Normal (Segmented) Transfer 1 = Expedited Transfer (see chapter 11.3.1)
s	Size indicator	0 = Data frame size is displayed 1 = Data frame size is not displayed
t	Toggle bit, toggled with each Segment change	0 = First and odd segments 1 = Second and even segments
c	Continue bit, marks following segments	0 = Further segments follow. 1 = This was the last segment.

The following sequence of telegrams results:

Initiate SDO Upload

		COB-ID	0	1	2	3	4	5	6	7
Request	Client → Server	0x600 + Node-ID	cs	Index		Subidx	Data			
			0x40	LSB	MSB		00	00	00	00
Response	Server → Client	0x580 + Node-ID	cs	Index		Subidx	Data			
			0x41	LSB	MSB		LSB	...	...	MSB

Segment Upload, first and odd segments

		COB-ID	0	1	2	3	4	5	6	7
Request	Client → Server	0x600 + Node-ID	cs	Data						
			0x60	00	00	00	00	00	00	00
Response	Server → Client	0x580 + Node-ID	cs	Data						
			0x00	LSB	...	...	...	...	...	MSB

Segment Upload, second and even segments

		COB-ID	0	1	2	3	4	5	6	7
Request	Client → Server	0x600 + Node-ID	cs	Data						
			0x70	00	00	00	00	00	00	00
Response	Server → Client	0x580 + Node-ID	cs	Data						
			0x10	LSB	...	...	...	...	...	MSB

Segment Upload, last segment

		COB-ID	0	1	2	3	4	5	6	7
Request	Client → Server	0x600 + Node-ID	cs	Data						
			0x60 or **0x70**	00	00	00	00	00	00	00
Response	Server → Client	0x580 + Node-ID	cs	Data						
			0xnn	LSB	...	...	...	...	...	MSB

11.3.4.2 Writing Segmented Transfer

The first telegram to write is executed via Command Specifier 0x21. The amount of entered data bytes in the data area defines the amount of data bytes to be transmitted in the following segment transfers. The following segments are controlled via Command Specifier 0x00 and 0x10 in toggling order until all data were transmitted. The last segment contains in the Command specifier (Bit 1...3) the amount of not used data bytes in the last telegram.
The resulting request and response telegrams are shown in the following sequence.
An example is described in chapter 12.3.1.4.

The Command Specifier have the following setup:

Initiate Download Command Specifier:

Request:	Bit	7	6	5	4	3	2	1	0
		ccs			0	n		e	s
Response:	Bit	7	6	5	4	3	2	1	0
		scs			0				

Download SDO Segment Command Specifier:

Request:	Bit	7	6	5	4	3	2	1	0
		ccs			t	n			c
Response:	Bit	7	6	5	4	3	2	1	0
		scs			t	0	0	0	0

Abbreviation	Description	Values
ccs	Client command Specifier	1 = Initiate download request 0 = Download sequence request
scs	Server command Specifier	3 = Initiate download request 1 = Download sequence response
n	Only valid if e =1 AND s = 1, in all other cases n = 0.	If valid: Amount of data bytes, that contain no useful data
e	Transfer type	0 = Normal Transfer 1 = Expedited Transfer (see chapter 11.3.2)
s	Size indicator	0 = Data frame size is displayed 1 = Data frame size is not displayed
t	Toggle bit, toggled with each Segment change	0 = First and odd segments 1 = Second and even segments
c	Continue bit, marks following segments	0 = Further segments follow. 1 = This was the last segment.

Initiate SDO Upload

	COB-ID	0	1	2	3	4	5	6	7
Request	Client →	0x600 +	cs	Index		Subidx	Data		
	Server	Node-ID	**0x21**	LSB	MSB		LSB	...	...	MSB
Response	Server →	0x580 +	cs	Index		Subidx	Data		
	Client	Node-ID	**0x41**	LSB	MSB		00	00	00	00

Segment Upload, first and odd segments

	COB-ID	0	1	2	3	4	5	6	7
Request	Client →	0x600 +	cs	Data					
	Server	Node-ID	**0x00**	00	00	00	00	00	00	00
Response	Server →	0x580 +	cs	Data					
	Client	Node-ID	**0x20**	00	00	00	00	00	00	00

Segment Upload, second and even segments

	COB-ID	0	1	2	3	4	5	6	7
Request	Client →	0x600 +	cs	Data					
	Server	Node-ID	**0x10**	00	00	00	00	00	00	00
Response	Server →	0x580 +	cs	Data					
	Client	Node-ID	**0x30**	00	00	00	00	00	00	00

Segment Upload, last segment

		COB-ID	0	1	2	3	4	5	6	7
Request	Client →	0x600 +	cs	Data						
	Server	Node-ID	**0xnn**	00	00	00	00	00	00	00
Response	Server →	0x580 +	cs	Data						
	Client	Node-ID	**0x10** oder **0x20**	LSB	...	...	...	...	...	MSB

11.4 PDO Function

The PDO (Process Data Objects) messages are messages with up to eight bytes of process data. The process data objects are mapped to the Rx/Tx-PDO's with the help of communication objects (communication/mapping parameter). Active Cube inverters support 3 RxPDO's (PLC ➔ inverter) and 3 TxPDO's (inverter ➔ PLC).

Process data objects are directly linked to application functions of the inverter.

PDO-message:

Byte	0	1	2	3	4	5	6	7
	data	data	data	data	data	data	data	data

The number of data bytes is 1 ... 8 and depends on the mapped objects. The byte alignment is in Intel format.

Byte	0	1	2	3	4	5
	16 bit object		32 bit object			
	LSB	MSB	LSB	...	...	MSB

Default Identifiers:

	Decimal	Hexadecimal
TxPDO1	384 + Node-ID	0x180 + Node-ID
RxPDO1	512 + Node-ID	0x200 + Node-ID
TxPDO2	640 + Node-ID	0x280 + Node-ID
RxPDO2	798 + Node-ID	0x300 + Node-ID
TxPDO3	896 + Node-ID	0x380 + Node-ID
RxPDO3	1024 + Node-ID	0x400 + Node-ID

11.5 Emergency Function

In the event of a communication error or an error inside the inverter, the inverter sends an emergency message. This emergency message includes the relevant error information. After error acknowledgement (fault reset), an emergency message is sent with all data bytes set to zero.

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x80 (=128)+ Node-ID	EEC	EEC	ER				MEC	MEC

EEC: Emergency Error Code according to DS301
ER: Emergency Register Code according to DS301
MEC: Manufacturer Error Code

The Manufacturer Error Code corresponds to the inverter Fault codes that are described in the Operating Instructions and in this documentation in chapter 16.4 "Fault messages".

Additional information is described in chapter 12.2.13 "0x1014/0 COB-ID Emergency Message".

11.6 SYNC

The SYNC message has two meanings.

The SYNC message is necessary for Rx/TxPDO with transmission type synchronous. The SYNC message synchronizes the different devices to communicate with data from the same (defined) time. As soon as the SYNC telegram is received, the data of all devices are “frozen” and then exchanged during the following data telegrams.

The RxPDO telegrams are collected until a SYNC telegram is received. With the reception of the SYNC telegram the data are transferred internally to the application parameters.

TxPDOs defined as synchronous send the actual application data on SYNC reception.

Additionally the SYNC mechanism can be used to synchronize the operating systems (OS) of different drives. This is useful when the electronic gear is used to enhance the performance of the application. The synchronization of the operating systems is described in chapter 11.10.

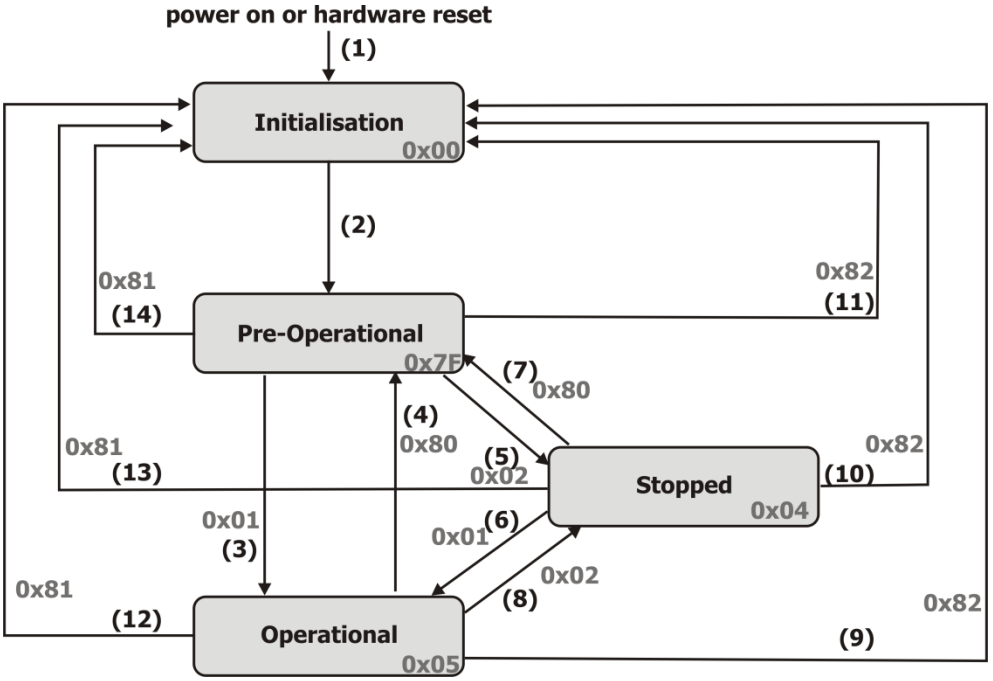
The SYNC message is a message with no data or with one byte data which is ignored. The default Identifier = 0x80 (=128).

COB-ID	Byte 0
0x80 (=128)	SYNC

11.7 NMT Functions

The NMT (Network management) functions describe the NMT state machine and NMT error control functions. The NMT state machine is controlled by NMT commands. The error control functions guarding and heartbeat are set up by associated communication objects and controlled by special protocols.
The NMT-State is displayed via the actual value parameter *Node-State* **1290**.

11.7.1 NMT state machine



A change of NMT-State may also be triggered by a communication (Bus-off, Guarding, etc.). The behaviour of the NMT state machine in such a case is described in Chapter 12.2.17 “0x1029/n Error Behavior”.

transition	NMT command
(1)	At power on NMT state Initialisation is entered autonomously
(2)	NMT state Initialisation finished → NMT state Pre-Operational entered automatically, device sends Boot-Up message
(3)	Start Remote Node
(4), (7)	Enter Pre-Operational
(5), (8)	Stop Remote Node
(6)	Start Remote Node
(9), (10), (11)	Reset Node. Communication objects 0x1nnn and application objects 0x6nnn are reset.
(12), (13), (14)	Reset Communication. Communication objects 0x1nnn are reset.

In state transition (2) Initialisation ➔ Pre-Operational the device sends the Boot-Up message.

11.7.2 Boot-Up message

Identifier	Byte 0
0x700 (=1792) + Node-ID	0

The Boot-Up message is sent automatically when the device is powered on or reset (i.e. fault reset). This helps the PLC recognizing to switch on a device (i.e. after a power failure and recovery) reliable during operation without Nodeguarding.

If the inverter is switched on after the PLC, the PLC can use this boot-up message to begin the initialization. The boot-up message signals the PLC, that the inverter is ready for the PLC to communicate. Using a NMT telegram “Reset Node” or “Reset Communication” forces a Reset of the node communication and results in a Boot-Up message.

Please refer also to chapter 11.8 “Guarding”.

11.7.3 NMT commands

	Byte 0	Byte 1
Identifier	Command Specifier	Node-ID
0	cs	id

id = 0 command addressed to **all** devices
id = 1...0x7F (=127) command addressed to device with Node-ID = id

cs:
1	Start Remote Node
2	Stop Remote Node
0x80 (=128)	Enter Pre-Operational
0x81 (=129)	Reset Node
0x82 (=130)	Reset Communication

NMT states and active communication objects:

	Pre-Operational	**Operational**	**Stopped**
PDO		**X**	
SDO	**X**	**X**	
SYNC	**X**	**X**	
Emergency	**X**	**X**	
Node control + NMT error control *****	**X**	**X**	**X**

***** NMT commands + Guarding/Heartbeat function

11.8 Guarding

Guarding response:
The inverter responds to every guarding request of the PLC. This is used by some PLCs when powering on to search for available devices. This response is done always independent of the settings of objects 0x100C/0 *Guard Time* and 0x100D/0 *Lifetime Factor.*

Guarding activation:
The Guarding is set whenever objects 0x100C/0 *Guard Time* and 0x100D/0 *Lifetime Factor* are both unequal to zero. The resulting guarding time is *Guard Time* **x** *Lifetime Factor*. Guarding is activated after setting the objects and on reception of the first guarding request.

Guarding fault behaviour:
If the inverter does not receive a guarding request within the specified guarding time a guarding event is triggered. The inverter's reaction to this guarding event is defined by objects 0x6007 *abort connection option code* and 0x1029 *error behaviour*.

Guarding sequence:
The PLC sends via a RTR (Remote Transmission Request) a guarding request with Identifier 0x700 (= 1792) + Node-ID (no data bytes). This remote frame is answered by the inverter with the same Identifier and one data byte. The data byte contains a toggle bit and the NMT state of the inverter.

PLC:

Identifier	
0x700 (=1792)+ Node-ID	RTR

Inverter:

	Byte 0							
Identifier	NMT state + toggle bit							
0x700 + Node-ID	7	6	5	4	3	2	1	0
	t	NMT state						

t:	Toggle bit toggled on each transmission (first transmission t = 0)	
NMT state:	0	Boot-Up
	4	Stopped
	5	Operational
	0x7F (=127)	Pre-Operational

11.9 Heartbeat

The heartbeat uses the producer/consumer method. The inverter as heartbeat consumer can monitor up to three heartbeat producers. The inverter can also send the heartbeat message (as heartbeat producer). The heartbeat contains the NMT state of the producer.

The heartbeat consumer function is set by object 0x1016/n *Consumer Heartbeat Time*. After setting the object the Monitoring of the heartbeat message(s) starts with reception of the first heartbeat message.
If the inverter does not receive a producer heartbeat message within the specified consumer heartbeat time, a heartbeat event is triggered. The reaction to this heartbeat event is defined by objects 0x6007 *abort connection option code* and 0x1029 *error behaviour*.

The heartbeat producer function is set by object 0x1017 *Producer Heartbeat Time*. If object 0x1017 *Producer Heartbeat Time* is set unequal to zero the inverter sends a heartbeat message periodically.

Heartbeat message:

Identifier	Byte 0							
	NMT state							
0x700 (=1792) + Node-ID	7	6	5	4	3	2	1	0
	r	NMT state						

r: reserved (always 0)

NMT state:
- 0 Boot-Up
- 4 Stopped
- 5 Operational
- 127 Pre-Operational

11.10 OS Synchronization

The operating System (OS) of the frequency inverter can be synchronized to the PLC or other devices. The synchronization of the OS enhances the performance of the complete plant. Synchronization is used to eliminate **phase** deviations of the CPUs between master and slave devices, so that calculations are done at the same time. Note, that only small deviations of the CPU clock frequencies between devices (i.e. different CPU Quartz cock frequencies) of ± 1 ‰ can be compensated. The synchronization time must be natural number as multiplier from 1 ms.

Synchronization via CANopen:
When using CANopen® without Systembus, the synchronization can be switched on and off. Synchronization can be done with CANopen® SYNC telegrams.

Synchronization via Systembus:
When using CANopen® simultaneously with Systembus, the synchronization can be set to either CANopen, Systembus or it can be switched off. Synchronization can be done with Systembus SYNC telegrams or Systembus RxPDO telegrams.

Note: When synchronizing the OS via CANopen®, the master has to support the synchronization mechanisms of CANopen®.

OS_SyncSource **1452**	
Operation mode	**Function**
0 - Auto	The synchronization source is selected automatically by the inverter.
1 - CANopen	The OS is synchronized via CANopen. **Factory setting**.
2 - Systembus	The OS is synchronized via Systembus.
99 - off	The OS is not synchronized with other devices.

Operation mode **Auto**: The selection is done via this decision table:

CANopen active	**Systembus active**	**Synchronization**
Yes	Yes	➔ Synchronization via CANopen
Yes	No	
No	Yes	➔ Synchronization via Systembus
No	No	➔ No Synchronization activated.

The CANopen "active status for synchronization" is recognized by the parameter setting **387** *CAN Node Number* >0 and a running synchronous PDO.
The Systembus "active status for synchronization" is recognized by the parameter setting **900** *Systembus Node ID* >0. Also parameter **1180** *Synchronization* has to be set to SYNC or an RxPDO.
The parameter **1451** *CANopen OS Synctime* can be used to shift the point of the synchronization inside of 1 ms. When you experience noises from a motor, shifting the *CANopen OS Synctime* can result in a better behaviour.

1453 *OS SyncSource Act* shows the active Synchronization source

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
1451	CANopen OS Synctime	700 us	900 us	800 us

Please refer to objects 0x1005 COB-ID SYNC object, 0x1006 Communication cycle period and 0x1007 Synchronous window length for CANopen Synchronization.

For the VPlus Scope Function the following sources are available for diagnosis:

Operation mode	Function
731 - B: Sync. OS <-> Sysbus Ok	1 = Synchronization OS to Systembus OK, 0 = Synchronization OS to Systembus not OK
852- SysBus SYNC time [us]	Shows the Synchronization cycle. Should show the set SYNC time or TxPDO time of the sending master.
853 SysBus SYNC position 1ms Task [us]	Shows the Synchronization time inside 1 ms. Should remain constant with small fluctuations.
854- B: Sync. OS <-> CANopen Ok	1 = Synchronization OS to CANopen OK, 0 = Synchronization OS to CANopen not OK
848- CANopen SYNC time [us]	Shows the Synchronization cycle. Should show the set SYNC time of object 0x1006.
849- CANopen SYNC position 1ms Task [us]	Shows the Synchronization time inside 1 ms. Should remain constant with small fluctuations.

11.11 Fault Reset

Depending on the settings and the operating status of the device a fault reset can be done like described:

- When using control via parameter *Local/Remote* **412** = Statemachine:
  Set bit 7 in 0x6040 Control word = 0x0080.
- Via the Stop key of the operator panel
  A reset via the STOP key can only be executed, if Parameter *Local/Remote* **412** allows the control via keypad

- via parameter *Error Acknowledgement* **103** which is assigned a logic signal or a digital input
  A reset via a digital input can only be executed, if Parameter *Local/Remote* **412** allows that control or if a physical input with the suffix (Hardware) is selected.

Some faults might re-occur after a fault reset. In these cases a certain action might be necessary (in example move away from a limit switch in the non-locked direction).

12 Objects

The available objects are marked via Index/Subindex and are to be addressed via this identification. This chapter describes all available objects.

12.1 Objects tabular overview

The objects are displayed in the next tables. The following definitions apply:

Access type			
Read only	The PLC is only allowed to read the data from the ACU.		
Read/write	The PLC is granted full access (read and write) to the ACU data.		
Data type			
Unsigned32	32 Bit value:	$0...2^{32}-1$ 0...0xFFFF FFFF	
Unsigned16	16 Bit value:	$0...2^{16}-1$ 0...0x FFFF	(0...65535)
Unsigned8	8 Bit value:	$0...2^{8}-1$ 0...0xFF	(0...255)
Integer32	Signed 32 Bit value:	$-2^{31}...2^{31}-1$ 0x8000 0000...0x7FFF FFFF	
Integer16	Signed 16 Bit value:	$2^{15}...2^{15}-1$ 0x8000...0x7FFF	(-32768...32767)
Integer8	Signed 8 Bit value: -	$2^{7}...2^{7}-1$ 0x80...0x7F	(-128...127)
Visible String	String up to 99 characters long. Transmission via Segmented Transfer.		
PDO mapping			
No	This object cannot be used for PDO exchange, only SDO is applicable.		
Tx	This object can be transmitted as PDO from ACU.		
Rx	This object can be transmitted as PDO to ACU.		

"Highest Sub-index supported" displays the highest Sub-index that is supported by this object.

12.1.1 Communication objects

Index	SubIndex	Name	SDO Access	Data type	PDO-mapping
0x1000	0	Device type	Read only	Unsigned32	No
0x1001	0	Error register	Read only	Unsigned8	No
0x1005	0	COB-ID SYNC object	Read/write	Unsigned32	No
0x1006	0	Communication cycle period	Read/write	Unsigned32	No
0x1007	0	Synchronous window length	Read/write	Unsigned32	No
0x1008	0	Manufacturer device name	Read only	Visible string	No
0x1009	0	Manufacturer hardware version	Read only	Visible string	No
0x100A	0	Manufacturer software version	Read only	Visible string	No
0x100C	0	Guard time	Read/write	Unsigned16	No
0x100D	0	Life time factor	Read/write	Unsigned8	No
0x1010		Store parameters			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	Save all parameters	Read/write	Unsigned32	No
	2	Save communication parameters	Read/write	Unsigned32	No
	3	Save application parameters	Read/write	Unsigned32	No

Index	SubIndex	Name	SDO Access	Data type	PDO-mapping
0x1011		Restore default parameters			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	Restore all default parame-ters	Read/write	Unsigned32	No
	2	Restore communication default parameters	Read/write	Unsigned32	No
	3	Restore application default parameters	Read/write	Unsigned32	No
0x1014	0	COB-ID emergency object	Read/write	Unsigned32	No
0x1016		Consumer heartbeat time			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	Consumer heartbeat time 1	Read/write	Unsigned32	No
	2	Consumer heartbeat time 2	Read/write	Unsigned32	No
	3	Consumer heartbeat time 3	Read/write	Unsigned32	No
0x1017	0	Producer heartbeat time	Read/write	Unsigned16	No
0x1018		Identity object			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	Vendor ID	Read only	Unsigned32	No
	2	Product code	Read only	Unsigned32	No
	3	Revision number	Read only	Unsigned32	No
	4	Serial number	Read only	Unsigned32	No
0x1029	0	Error behaviour	Read only	Unsigned8	No
	1	Communication error	Read/write	Unsigned8	No
0x1200	0	Server SDO parameter	Read only	Unsigned8	
	1	COB-ID Rx	Read only	Unsigned32	No
	2	COB-ID Tx	Read only	Unsigned32	No
0x1400		RxPDO1 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1401		RxPDO2 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1402		RxPDO3 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1600		RxPDO1 mapping parameter			
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No

0x1601		RxPDO2 mapping parameter			
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No
0x1602		RxPDO3 mapping parameter			
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No
0x1800		TxPDO1 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1801		TxPDO1 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1802		TxPDO1 communication parameter			
	0	Highest sub-index supported	Read only	Unsigned8	No
	1	COB-ID	Read/write	Unsigned32	No
	2	Transmission type	Read/write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/write	Unsigned16	No
0x1A00		TxPDO1 mapping parameter			
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No
0x1A01		TxPDO2 mapping parameter			
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No

0x1A02	TxPDO3 mapping parameter				
	0	No. of mapped objects	Read/write	Unsigned8	No
	1	1. mapped obj.	Read/write	Unsigned32	No
	2	2. mapped obj.	Read/write	Unsigned32	No
	3	3. mapped obj.	Read/write	Unsigned32	No
	4	4. mapped obj.	Read/write	Unsigned32	No
	5	5. mapped obj.	Read/write	Unsigned32	No
	6	6. mapped obj.	Read/write	Unsigned32	No
	7	7. mapped obj.	Read/write	Unsigned32	No
	8	8. mapped obj.	Read/write	Unsigned32	No

12.1.2 Manufacturer objects

Index	Sub-index	Designation	SDO Access	Data type	PDO-mapping	Factory setting	Min...Max	Belonging. Param.
0x2nnn	0, 1, ... 9	Manufacturer specific Direct access to inverter parameters Read/write access by SDO transfer only Please refer to chapter 12.3.1 „Handling of data sets/cyclic writing of the parameters“ and 11.3.2 „Write Access”						
0x3000	0	Sync Jitter	Read/write	Unsigned16	Nein			
0x3001	0	Digital In actual values	Read only	Unsigned16	Tx	-	-	-
0x3002	0	Digital Out actual values	Read only	Unsigned16	Tx	-	-	-
0x3003	0	Digital Out set values	Read/write	Unsigned16	Rx	0	0...0x1F	
0x3004	0	Boolean Mux	Read only	Unsigned16	Tx	-	-	-
0x3005	0	Boolean Demux	Read/write	Unsigned16	Rx	0	0...0xFFFF	-
0x3006	0	Percentage set value	Read/write	Unsigned16	Rx	0	0x8AD0... 0x7530	-
0x3007	0	Percentage actual value 1	Read only	Unsigned16	Tx	-	-	-
0x3008	0	Percentage actual value 2	Read only	Unsigned16	Tx	-	-	-
0x3011	0	Act. value Word 1	Read only	Unsigned16	Rx	-	-	-
0x3012	0	Act. value Word 2	Read only	Unsigned16	Rx	-	-	-
0x3021	0	Act. value Long 1	Read only	Unsigned32	Rx	-	-	-
0x3022	0	Act. value Long 2	Read only	Unsigned32	Rx	-	-	-
0x3111	0	Ref. value Word 1	Read/write	Unsigned16	Tx	0	0...0xFFFF	-
0x3112	0	Ref. value Word 2	Read/write	Unsigned16	Tx	0	0...0xFFFF	-
0x3121	0	Ref. value Long 1	Read/write	Unsigned32	Tx	0	0... 0xFFFF.FFFF	-
0x3122	0	Ref. value Long 2	Read/write	Unsigned32	Tx	0	0... 0xFFFF.FFFF	-
0x5F10		Gear factor [g)]						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Numerator	Read/write	Integer16	Rx	1	1...0x7FFF	p.1123
	2	Denominator	Read/write	Unsigned16	Rx	1	1...0xFFFF	p.1124
	3	Resync on Change	Read/write	Integer16	No	1	0...1	p.1142
0x5F11		Phasing 1 [g)]						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Offset	Read/write	Integer32	No	0x0001.0000	0x8000.0000 ... 0x7FFF.FFFF	p.1125 DS1
	2	Speed	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1126 DS1
	3	Acceleration	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1127 DS1

Index	Sub-index	Designation	SDO Access	Data type	PDO-mapping	Factory setting	Min...Max	Belonging. Param.
0x5F12		Phasing 2 g)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Offset	Read/write	Integer32	No	0x0001.0000	0x8000.0000 ... 0x7FFF.FFFF	p.1125 DS2
	2	Speed	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1126 DS2
	3	Acceleration	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1127 DS2
0x5F13		Phasing 3 g)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Offset	Read/write	Integer32	No	0x0001.0000	0x8000.0000 ... 0x7FFF.FFFF	p.1125 DS3
	2	Speed	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1126 DS3
	3	Acceleration	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1127 DS3
0x5F14		Phasing 4 g)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Offset	Read/write	Integer32	No	0x0001.0000	0x8000.0000 ... 0x7FFF.FFFF	p.1125 DS4
	2	Speed	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1126 DS4
	3	Acceleration	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1127 DS4
0x5F15	0	In Gear Threshold	Read/write	Unsigned32	No	0	0... 0x7FFF.FFFF	p.1168
0x5F16	0	In Gear Time	Read/write	Unsigned16	No	10	1...0xFFFF	p.1169
0x5F17		Position Controller v) u) h) i) p) t) g)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Time Constant [ms]	Read/write	Integer32	No	10,00	1,00...300,00	p.1104
	2	Limitation	Read/write	Unsigned32	No	327680	0... 0x7FFF.FFFF	p.1118
0x5F18	0	Master Synchronization Offset g)	Read/write	Integer32	No	0	0x8000.0000 ... 0x7FFF.FFFF	p.1284
0x5FF0	0	Active motion block t)	Read only	Unsigned8	Tx	-	-	-
0x5FF1	0	Motion block to resume t)	Read only	Unsigned8	Tx	-	-	-

v) Velocity Mode only: This Object is only used in Velocity mode [rpm].
u) Profile Velocity Mode only: This Object is only used in Profile Velocity mode [u/s].
h) Homing Mode only: This Object is only used in Homing mode.
i) Interpolated Position Mode only: This Object is only used in Interpolated Position mode.
p) Profile Position Mode only: This Object is only used in Profile Position mode.
g) Electronic gear: slave Mode only: This Object is only used in Electronic Gear mode.
t) Table travel record mode only: This Object is only used in Table travel record mode.

12.1.3 Device profile objects

Index	Sub-index	Designation	SDO Access	Data type	PDO-mapping	Factory setting	Min...Max	Belonging. Param.
0x6007	0	Abort connection option code	Read/write	Integer16	No	1	-2...3	p.388
0x603F	0	Error code	Read only	Unsigned16	No	-	-	-
0x6040	0	Control word	Read/write	Unsigned16	Rx	-	-	p.410
0x6041	0	Status word	Read/only	Unsigned16	Tx	-	-	p.411
0x6042	0	Target velocity v)	Read/write	Interger16	Rx	0	-32768... 32767	-
0x6043	0	Target velocity demand	Read only	Integer16	Tx	-	-	-
0x6044	0	Control effort	Read only	Integer16	Tx	-	-	-
0x6046		Velocity min max						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Velocity min amount	Read/write	Unsigned32	No	0	0...32767	p.418
	2	Velocity max amount	Read/write	Unsigned32	No	32767	0...32767	p.419
0x6048		Velocity acceleration v)				-	-	-
	0	Highest sub-index supported	Read only	Unsigned8	No			
	1	Delta speed	Read/write	Unsigned32	No	150	1...32767	p.420 &
	2	Delta time	Read/write	Unsigned16	No	1	1...65535	p.422
0x6049		Velocity deceleration v)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Delta speed	Read/write	Unsigned32	No	150	1...32767	p.421 &
	2	Delta time	Read/write	Unsigned16	No	1	1...65535	p.423
0x604A		Velocity quick stop v)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Delta speed	Read/write	Unsigned32	No	150	1...32767	p.424 &
	2	Delta time	Read/write	Unsigned16	No	1	1...65535	p.425
0x6060	0	Modes of operation	Write only	Integer8	Rx	2	-3...9	-
0x6061	0	Modes of operation display	Read only	Integer8	Tx	2	-	-
0x6064	0	Position actual value	Read only	Integer32	Tx	-	0x8000.0000 ... 0x7FFF.FFFF	p.1108

Index	Sub-index	Designation	SDO Access	Data type	PDO-mapping	Factory setting	Min...Max	Belonging. Param.
0x6065	0	Following error window	Read/write	Unsigned32	No	0xFFFF.FFFF	0... 0xFFFF.FFFF	p.1105
0x6066	0	Following error time out	Read/write	Unsigned16	No	10	0...65535	p.1119
0x6067	0	Position window	Read/write	Unsigned32	No	0xFFFF.FFFF	0... 0xFFFF.FFFF	p.1165
0x6068	0	Position window time	Read/write	Unsigned16	No	10	0...65535	p.1166
0x606C	0	Velocity Actual value u)	Read	Integer32	Tx			-
0x606D	0	Velocity Window u)	Read/write	Unsigned16	No	1000	0...65535	p.1276
0x606E	0	Velocity Window Time u)	Read/write	Unsigned16	No	0	0...65535	p.1277
0x606F	0	Velocity Threshold u)	Read/write	Unsigned16	No	100	0...65535	p.1278
0x6070	0	Velocity Threshold Time u)	Read/write	Unsigned16	No	0	0...65535	p.1279
0x6071	0	Target torque	Read/write	Integer16	Rx			-
0x6077	0	Torque actual value	Read only	Integer16	Tx			p.224
0x6078	0	Current actual value	Read only	Integer16	Tx			p.214
0x6079	0	DClink circuit voltage	Read only	Integer32	Tx			p.222
0x607A	0	Target position p)sv)	Read/write	Integer32	Rx	0	0x8000.0000 ... 0x7FFF.FFFF	p.1202
0x607C	0	Home offset h)	Read/write	Integer32	No	0	0x8000.0000 ... 0x7FFF.FFFF	p.1131
0x6081	0	Profile velocity p) i) u)	Read/write	Unsigned32	Rx	0x0005.0000	1... 0x7FFF.FFFF	
0x6083	0	Profile acceleration p) i) u)	Read/write	Unsigned32	Rx	0x0005.0000	1... 0x7FFF.FFFF	
0x6084	0	Profile deceleration p) i) u)	Read/write	Unsigned32	Rx	0x0005.0000	1... 0x7FFF.FFFF	
0x6085	0	Quick stop deceleration h) i) p) t) u)	Read/write	Unsigned32	No	0x000A.0000	1... 0x7FFF.FFFF	p.1179
0x6086	0	Motion profile type u)	Read/write	Integer16	No	3	0...3	-
0x6091		Gear ratio						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Motor revolutions	Read/write	Unsigned32	No	1	1...65535	p.1116
	2	(Driving) Shaft revolutions	Read/write	Unsigned32	No	1	1...65535	p.1117
0x6092		Feed constant						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Feed	Read/write	Unsigned32	No	0x0001.0000	1... 0x7FFF.FFFF	p.1115
	2	(Driving) Shaft revolutions	Read/write	Unsigned32	No	1	1	
0x6098	0	Homing method h)	Read/write	Integer8	No	0	0...35	p.1130
0x6099		Homing speeds h) l)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Speed during search for switch	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1132
	2	Speed during search for zero	Read/write	Unsigned32	No	0x0002.0000	1... 0x7FFF.FFFF	p.1133

Index	Sub-index	Designation	SDO Access	Data type	PDO-mapping	Factory setting	Min...Max	Belonging. Param.
0x609A	0	Homing acceleration h)	Read/write	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF	p.1134
0x60C1		Interpolation data record i)						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Interpolation data record 1	Read/write	Integer32	Rx	0	0x8000.0000 ... 0x7FFF.FFFF	
0x60F4	0	Following error actual value	Read only	Integer32	Tx			p.1109
0x60F8	0	Max Slippage u)	Read/write	Integer32	No	0		p. 1275
0x60FF	0	Target velocity u)sv)	Read/write	Integer32	Rx			

v) Velocity Mode: This Object is used in Velocity mode [rpm].
u) Profile Velocity Mode: This Object is used in Profile Velocity mode [u/s].
h) Homing Mode: This Object is used in Homing mode.
i) Interpolated Position Mode: This Object is used in Interpolated Position mode.
p) Profile Position Mode: This Object is used in Profile Position mode.
t) Table travel record mode: This Object is used in Table travel record mode.
l) Move away from limit switch Mode: This Object is used in Move away from limit switch mode.
sp) Cyclic Sync Position mode: This Object is used in Cyclic Synchronous Position mode.
sv) Cyclic Sync Velocity mode: This Object is used in Cyclic Synchronous Velocity mode.

The Modes “Homing”, “Interpolated Position”, “Profile Position”, “Profile Velocity” and “Table travel record”, “Move away from Limit Switch”, Cyclic Sync Position” and “Cyclic Sync Velocity” require a configuration capable of Positioning. Check chapter 14.4 “Motion control configurations” for details.

The notations of CANopen® objects and parameters can be different (refer to the corresponding object description).

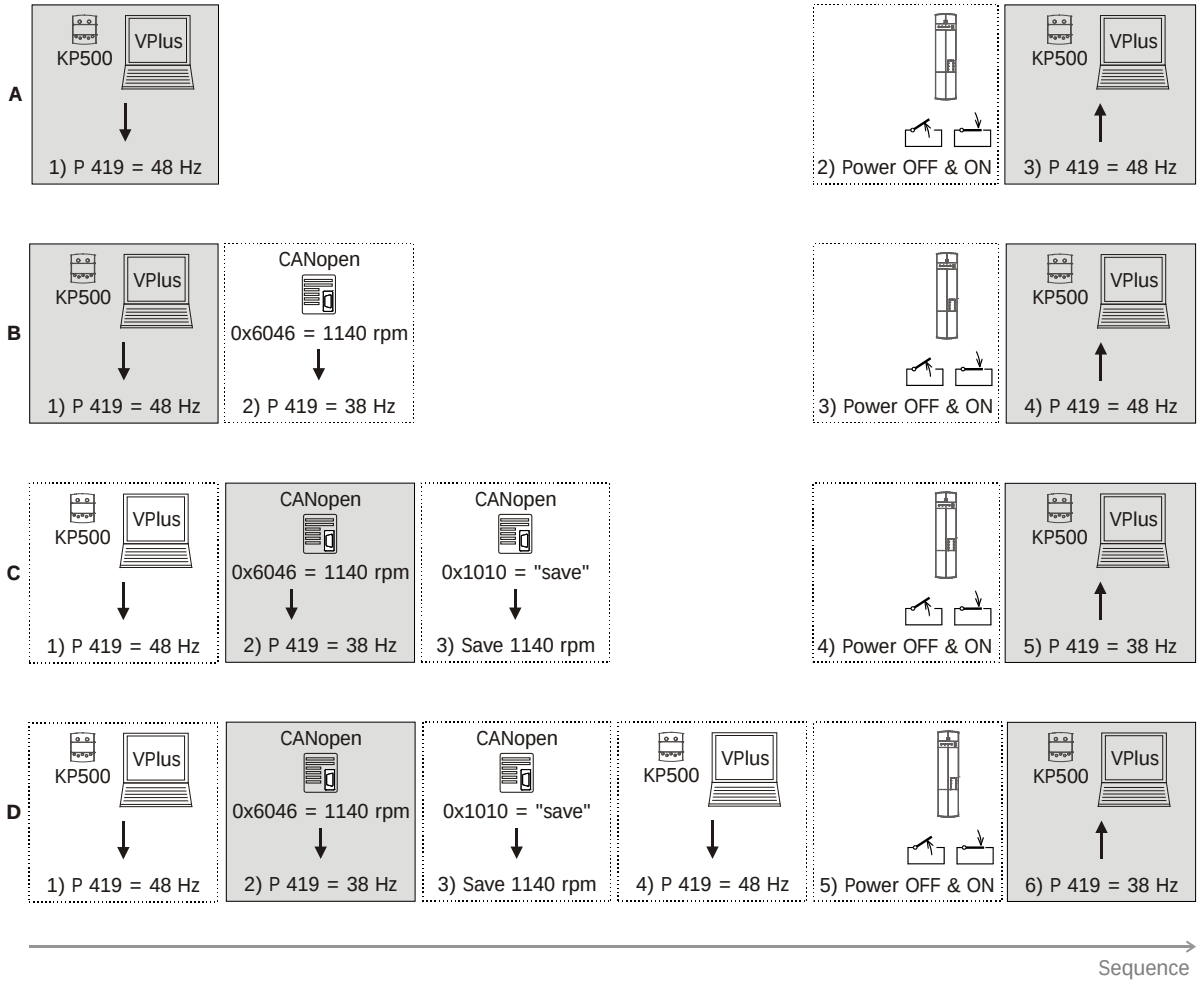
Note

Some of the above listed CANopen® objects have corresponding inverter parameters.

These objects are handled in a special way. If one of these CANopen® objects has been written by SDO followed by a “save” command (see object 0x1010), the written value is stored to non-volatile memory of the inverter. After the next power on of the inverter these CANopen® object values are restored again and overwrite the inverter parameter values.

Be careful when using this method. If a CANopen® object was written and saved and then the corresponding inverter parameter was set by e. g. VPlus, the next power on cycle overwrites the value set by VPlus with the value stored by the “save” command.

Effect of the “save” command (Object 0x1010)
(sequences of writing parameters and objects, examples)



A Value of a parameter is set via KP500 or VPlus. No “save” command.

1) Setting of *Maximum Frequency* **419** = 48 Hz at KP500 or in VPlus.
2) Power OFF and ON.
3) The value of KP500/VPlus is active (48 Hz).

B No “save” command. The value of the CANopen® object is overwritten.

1) Setting of *Maximum Frequency* **419** = 48 Hz at KP500 or in VPlus.
2) Setting of CANopen® object 0x6046 = 1140 rpm* (equivalent to 38 Hz).
3) Power OFF and ON.
4) Parameter value of KP500/VPlus overwrites the value of the CANopen® object. The value of KP500/VPlus is active (48 Hz).

C “Save” command. The value of the CANopen® object is stored.

1) Setting of *Maximum Frequency* **419** = 48 Hz at KP500 or in VPlus.
2) Setting of CANopen® object 0x6046 = 1140 rpm* (equivalent to 38 Hz).
3) “Save” command via CANopen® object 0x1010.
4) Power OFF and ON.
5) The value of CANopen® object 0x6046 is active (38 Hz).

D “Save” command. The value of the CANopen® object is stored – even if the corresponding parameter value has been changed after the “save” command.

1) Setting of *Maximum Frequency* **419** = 48 Hz at KP500 or in VPlus.
2) Setting of CANopen® object 0x6046 = 1140 rpm* (equivalent to 38 Hz).
3) “Save” command via CANopen® object 0x1010.
4) Setting of *Maximum Frequency* **419** = 48 Hz at KP500 or in VPlus.
5) Power OFF and ON.
6) Value of CANopen® object 0x6046 overwrites the parameter value. The value of CANopen® object 0x6046 is active (38 Hz).

* Internal conversion to a frequency value taking into account the *No. of Pole Pairs* **373**. In this example the number of pole pairs is two (four-pole machine).

NOTE

There are inverter parameters calculated from CANopen® objects which require the no. of pole pairs for calculating the corresponding value for inverter parameters (e. g. deceleration or acceleration parameters). These calculations always use the no. of pole pairs from data set 1. If the no. of pole pairs is different in the data sets, the result of this operation will not be clear for the user. To avoid confusion it is recommended to write the inverter parameters via the SDO channel using the 0x2nnn (manufacturer) objects and not to use the CANopen® objects. This way, inconsistencies are avoided.

All CANopen® DS402 objects with corresponding inverter parameters are pointed out in this manual.

12.2 Communication Objects (0x1nnn)

The communication objects 0x1nnn contain all parameters for the communication.

For easier usage, the objects are summarized by a table in each paragraph. This table is marked additional by colour.

Orange colour	= Read Only object
Green colour	= Read and Write object
Blue colour	= Write only object

Used abbreviations:

Access: Access type
r/w: Read/Write
ro: Read only
wo: Write only

Map: Mapping
Def.-Val: Default value of object

The examples show some typical data telegrams that can be watched or used with a CAN analysis tool. The order of the examples consider the standard CANopen® Format: Lowest Byte left, Highest Byte right.

The headings are displayed in the format *Index/Subindex Objectname*.

12.2.1 0x1000/0 Device Type

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1000	0	Device Type	Unsigned 32	ro	No	0

The device identification is carried out during the start of the network. The information on the device type and the functionality (type) are prescribed by the CANopen® standards.

Object 0x1000/0			
Additional Information		Device Profile Number	
Mode Bits		Type	
31	24	23 16	15 0

The "Drives and Motion Control" standard device profile used by the frequency inverter is portrayed as device profile number 402. The additional information specifies the device functionality of the frequency inverter.

Device Profile Number	= 402	drives and motion control
Type	= 42	servo drive
Mode bits	= 0	unused

Up to Firmware 5.2.0 (including) “Type” depends on the setting of parameter *Configuration* **30**.
A motion control configuration (*Configuration* **30**=x**40**) sets type = 42 “servo drive”.
Other configurations set type = 41 “frequency converter”.

Firmware 5.3.0 always sets type = 42 “servo drive”.

Example:

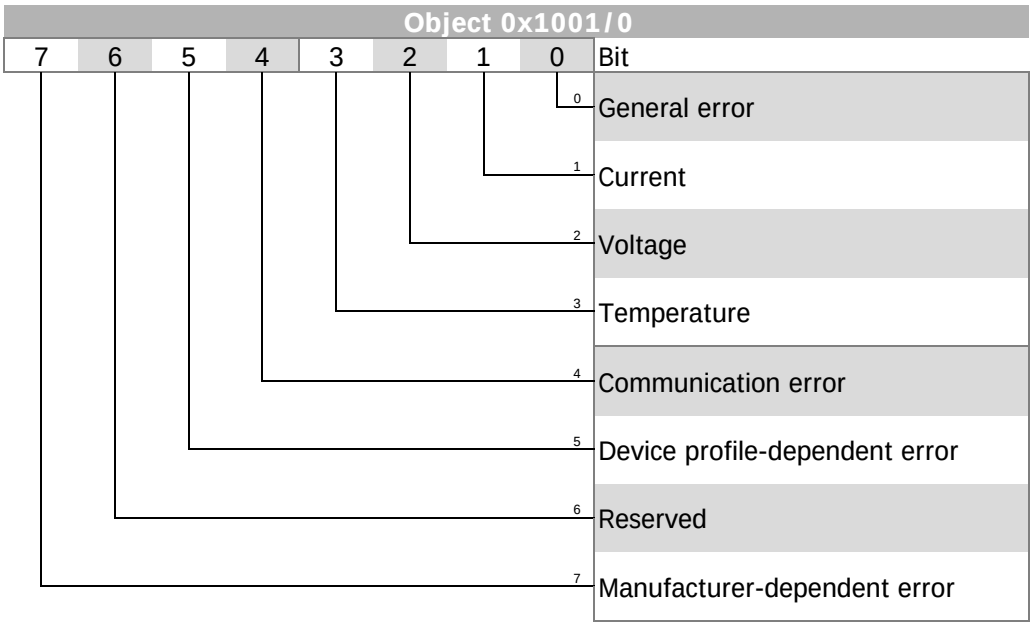
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 10	00	00 00 00 00
Reply	581	43	00 10	00	92 01 42 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.2 0x1001/0 Error Register

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1001	0	Error Register	Unsigned 8	ro	No	0

Object 0x1001/0 is the error register for internal errors of the frequency inverter. The status error-free (0x1001/0 = 0) or error exists (0x1001/0 ≠ 0) is displayed.
Detailed information about the current device fault can be checked VPlus with parameter *Actual Error* **259** and **260** via CANopen (see chapter 16.4 “Fault messages”).
Also the emergency message contains additional information, which can be evaluated by the PLC (see chapters 11.5 “Emergency Function” and 12.5.2 “0x603F/0 Error code”.



Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	01 10	00	00 00 00 00
Reply	581	4F	01 10	00	00 01 41 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.3 0x1005/0 COB-ID SYNC Message

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1005	0	COB-ID SYNC Message	Unsigned 32	r/w	No	0

Object 0x1005 *COB-ID SYNC message* defines the identifier for the SYNC message as well as whether the CANopen® device generates the SYNC.

The default value of this object is 128 (identifier = 128, SYNC message not generated).

Object 0x1005/0				
Bit 31	Bit 30	Bit 29	Bit 11 ... 28	Bit 0 ... 10
X	gen	frame	**0**	11 bit CAN-ID

Bit 31: X = don't care

Bit 30: 0 = SYNC message not generated
1 = SYNC message generated

Bit 29: 0 = 11 bit ID
1 = 29 bit ID **NOT ALLOWED**

Bit 0 ... 10: 11 bit CAN-ID

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	05 10	00	00 00 00 00
Reply	581	43	05 10	00	80 00 00 00
Write Access	601	23	05 10	00	81 00 00 00
Reply	581	60	05 10	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.4 0x1006/0 Communication Cycle Period

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1006	0	Communication Cycle Period	Unsigned 32	r/w	No	0

The *communication cycle period* is the time distance between two consecutive SYNC messages. The SYNC message is used by the inverter for synchronisation of the motion control system to the SYNC message. This is especially important for the interpolated position mode.

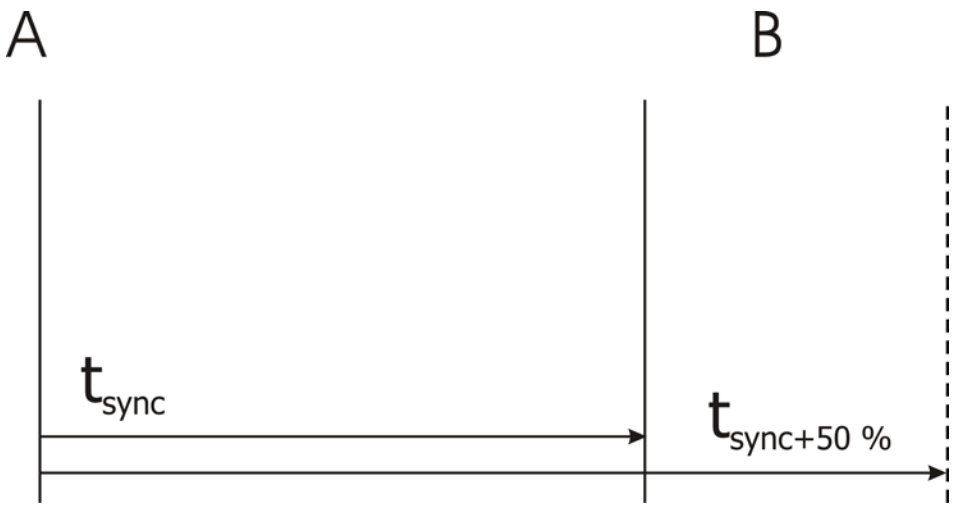
The value for *communication cycle period* is given in multiples of micro seconds.
Values smaller than 20000 (20 ms) are allowed.

The synchronization of the inverter to an external clock has to be met under the condition, that at least one RxPDO or TxPDO is defined as synchronous object and is activated. The definitions of the TxPDO / RxPDO objects can be changed via objects 0x1400 / 0x1800.

The inverter can only process the SYNC mechanism in multiples of milliseconds. For this reason the allowed values for object 0x1006/0 *communication cycle period* are multiples of milliseconds.

E.g.: 0x1006/0 = 4000 = 4 ms

If the *communication cycle period* is NOT set (0x1006/0 = 0), the inverter measures the time distance between the SYNC messages over the first 11 messages. Please note, that the monitoring function is deactivated for setting “0”. The measurement is solely for internal uses of the frequency inverter. The time must not change after the measurement.



The time distance between two consecutive SYNC messages is monitored.
If object 0x1006/0 *communication cycle period* is set to a value other than zero, then a communication error event is triggered whenever the time defined by 0x1006/0 is exceeded by more than 50%.

After SYNC telegram “A”, SYNC telegram “B” has to be received latest after the set SYNC time + 50 %.

If object 0x1006/0 *communication cycle period* is not set (= zero), then this monitoring function is **not** active.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	06 10	00	00 00 00 00
Reply	581	43	06 10	00	00 00 00 00
Write Access	601	23	06 10	00	A0 0F 00 00
Reply	581	60	06 10	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.5 0x1007/0 Synchronous window length

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1007	0	Synchronous window length	Unsigned 32	r/w	No	See Text

Synchronous window length is the time span after a SYNC message in which the inverter is supposed to update its data from receive PDOs and to send transmit PDOs. If either of these actions is not possible in the specified time an emergency message is sent and all remaining synchronous PDOs are discarded until the next SYNC message.

The value for *synchronous window length* is given in multiples of micro seconds.
E.g.: 0x1007/0 = 2000 = 2 ms

If object 0x1007/0 *synchronous window length* is not set (= zero), then this monitoring function is **not** active.

To avoid unnecessary bus load, the emergency message is sent once only. The next emergency message concerning this problem will be sent after the successful processing of all synchronous PDOs within the *synchronous window length* and a new violation of *synchronous window length*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	07 10	00	00 00 00 00
Reply	581	43	07 10	00	00 00 00 00
Write Access	601	23	07 10	00	D0 07 00 00
Reply	581	60	07 10	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.6 0x1008/0 Manufacturer Device Name

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1008	0	Manufacturer Device name	Visible string	ro	No	See Text

The device name is displayed as a sequence of ASCII characters.
Example : "**ACTIVE CUBE**"

The object 0x1008/0 supports the segmented SDO transfer. Check for descriptions and examples chapter 11.3.4.1 and 12.3.1.5.

12.2.7 0x1009/0 Manufacturer Hardware Version

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1009	0	Manufacturer Hardware version	Visible string	ro	No	See Text

The device version is displayed as a sequence of ASCII characters.
Example : "**ACU 400 512 344**"

The object 0x1009/0 supports the segmented SDO transfer. Check for descriptions and examples chapter 11.3.4.1 and 12.3.1.5.

12.2.8 0x100A/0 Manufacturer Software Version

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x100A	0	Manufacturer Software version	Visible string	ro	No	See Text

The software version is displayed as a sequence of ASCII characters.
Example : "**5.4.0**"

The object 0x100A/0 supports the segmented SDO transfer. Check for descriptions and examples chapter 11.3.4.1 and 12.3.1.5.

12.2.9 0x100C/0 Guard Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x100C	0	Guard time	Unsigned 16	r/w	No	0

The response monitoring time is calculated by the multiplication of the objects g*uard time* and *lifetime factor*. Object 0x100C/0 defines the *guard time* in units of one millisecond. *Guard time* = 0 deactivates the guarding function.
If the response monitoring time is exceeded, the node reacts as defined by the setting of object 0x6007 *abort connection option code*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	0C 10	00	00 00
Reply	581	4B	0C 10	00	00 00
Write Access	601	2B	0C 10	00	D0 07
Reply	581	60	0C 10	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.10 0x100D/0 Lifetime Factor

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x100D	0	Lifetime factor	Unsigned 8	r/w	No	0

The object "Lifetime Factor" is the multiplier for *guard time. Lifetime factor* = 0 deactivates the guarding function.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	0D 10	00	00
Reply	581	4F	0D 10	00	00
Write Access	601	2F	0D 10	00	05
Reply	581	60	0D 10	00	00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.11 0x1010/n Store Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1010	0	Highest sub-index supported	Unsigned8	ro	No	3
	1	Store all parameters	Unsigned32	r/w	No	See text
	2	Store communication parameters	Unsigned32	r/w	No	See text
	3	Store application parameters	Unsigned32	r/w	No	See text

With object 0x1010/n parameter/object settings can be stored to non-volatile memory. This object supports 3 sub-indexes with different functions.

Writing “save” to 0x1010/3 stores all application parameters (0x6nnn) to non-volatile memory.

Specification of write “save” command

LSB			MSB
“**s**”	“**a**”	“**v**”	“**e**”
0x73	0x61	0x76	0x65

Writing a value other than "save" results in an SDO abort. The store command is **not** processed.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	10 10	01	00 00 00 00
Reply	581	43	10 10	01	01 00 00 00
Write Access	601	23	10 10	01	73 61 76 65
Reply	581	60	10 10	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.12 0x1011/n Restore default Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1011	0	Highest sub-index supported	Unsigned8	ro	No	3
	1	Restore all parameters	Unsigned32	r/w	No	See text
	2	Restore communication parameters	Unsigned32	r/w	No	See text
	3	Restore application parameters	Unsigned32	r/w	No	See text

With object 0x1011/n parameters/objects can be set to their default values. This object supports 3 sub-indexes with different functions.

Writing “load” to 0x1011/3 restores all application parameters (0x6nnn).

Specification of write “load” command

LSB			MSB
“**l**”	“**o**”	“**a**”	“**d**”
0x6C	0x6F	0x61	0x64

Writing a value other than "load" results in an SDO abort. The restore defaults command is **not** processed.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	11 10	01	00 00 00 00
Reply	581	43	11 10	01	01 00 00 00
Write Access	601	23	11 10	01	6C 6F 61 64
Reply	581	60	11 10	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.13 0x1014/0 COB-ID Emergency Message

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1014	0	COB-ID Emergency Message	Unsigned32	r/w	No	See text

The identifier and thus the definition of the priority for the emergency message can be set with object 0x1014/0.

The default value of the identifier is 128 + Node ID (valid).

Object 0x1014/0				
Bit 31	Bit 30	Bit 29	Bit 11 ... 28	Bit 0 ... 10
valid	**0**	frame	**0**	11 bit CAN-ID

Bit 31: 0 = EMCY existent / valid
1 = EMCY non-existent / not valid

Bit 29: 0 = 11 Bit ID
1 = 29 Bit ID **NOT ALLOWED**

Bit 0 ... 10: 11 bit CAN-ID

The emergency message is transmitted with the emergency message COB-ID and comprises eight bytes. This object is generated in individual cases and the fault acknowledgement signaled by an emergency message with the data contents equal to zero. The contents are coded according to the following table:

Emergency Message	
Byte	**Contents**
0	Low-byte error code (0x603F)
1	High-byte error code (0x603F)
2	Error register (0x1001)
3	**0**
4	**0**
5	**0**
6	Low-byte, internal error code
7	High-byte, internal error code

Bytes 0, 1 and 2 have a fixed definition within the emergency object. Bytes 6 and 7 are used product-specifically on the basis of the specification.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	14 10	00	00 00 00 00
Reply	581	43	14 10	00	81 00 00 00
Write Access	601	23	14 10	00	81 00 00 00
Reply	581	60	14 10	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.14 0x1016/n Consumer Heartbeat Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1016	0	Highest sub-index supported	Unsigned8	ro	No	3
	1	Consumer Heartbeat Time 1	Unsigned32	r/w	No	See text
	2	Consumer Heartbeat Time 2	Unsigned32	r/w	No	See text
	3	Consumer Heartbeat Time 3	Unsigned32	r/w	No	See text

Up to three heartbeat producers can be monitored with object 0x1016/n (controlled via sub-indexes n = 1 ... 3). Setting "Consumer Heartbeat Time" = 0 means no monitoring.
Node ID identifies the device to be monitored. The *Heartbeat Time* states the maximum time in milliseconds between two heartbeat messages of the heartbeat producer to be monitored. If this time is exceeded, the monitoring node reacts as defined by the setting of object 0x6007 *abort connection option code*.

Value of the consumer heartbeat time		
Bit 24 to Bit 31	Bit 16 to Bit 23	Bits 0 to Bit 15
not used	Node ID	Heartbeat Time

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	16 10	01	00 00 00 00
Reply	581	43	16 10	01	02 00 00 00
Write Access	601	23	16 10	01	20 00 03 00
Reply	581	60	16 10	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.15 0x1017/0 Producer Heartbeat Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1017	0	Producer Heartbeat Time	Unsigned16	r/w	No	0 ms

The time for the transmission of a heartbeat object is set with object 0x1017/0. The setting “Producer Heartbeat Time” = 0 means that no heartbeat object is transmitted.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	17 10	00	00 00
Reply	581	4B	17 10	00	00 00
Write Access	601	23	17 10	00	20 00
Reply	581	60	17 10	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.16 0x1018/n Identity Object

The identity object provides information on the device manufacturer as well as the device itself.

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1018	0	Highest sub-index supported	Unsigned8	ro	No	4
	1	Vendor ID	Unsigned32	ro	No	See text
	2	Product code	Unsigned32	ro	No	See text
	3	Revision number	Unsigned32	ro	No	See text
	4	Serial number	Unsigned32	ro	No	See text

The vendor ID "**0xD5**" identifies the manufacturer **Bonfiglioli Vectron GmbH**. This vendor ID has been assigned by the CANopen® users' organization "CAN in Automation" (CiA®) in Erlangen/Germany (*www.can-cia.org*).

Product code: displays the inverter's type code.
Revision number: displays the inverter's EtherCAT®/CANopen® system revision.
Serial number: displays the inverter's serial number.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	18 10	01	00 00 00 00
Reply	581	43	18 10	01	05 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.17 0x1029/n Error Behavior

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1029	0	Highest sub-index supported	Unsigned8	ro	No	1
	1	Communication error	Unsigned8	r/w	No	0

The Error Behavior object defines the behavior of the NMT state machine in the event of a communication error (BusOff, Guarding, Heartbeat, SYNC, RxPDO-length).

Value	Function
0	Change to NMT state Pre-Operational (default) (only if currently in NMT state Operational)
1	No change of NMT state
2	Change to NMT state Stopped

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	29 10	01	00 00 00 00
Reply	581	43	29 10	01	05 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.18 0x1200/n SDO Server Parameter

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1200	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	COB-ID client → server (Rx)	Unsigned32	ro	No	See text
	2	COB-ID server → client (Tx)	Unsigned32	ro	No	See text

Object 0x1200 defines the SDO server parameters. The values are read-only and pre defined according to the device node address.

COB-ID client → server (Rx) = 1536 + node address

COB-ID server → client (Tx) = 1408 + node address

Object 0x1200/1, 2				
Bit 31	Bit 30	Bit 29	Bit 11 ... 28	Bit 0 ... 10
valid	**0**	frame	**0**	11 bit CAN-ID

Bit 31: **0** = SDO existent / valid

Bit 29: **0** = 11 Bit ID

Bit 0 ... 10: 11 bit CAN-ID

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	00 12	02	00 00 00 00
Reply	581	43	00 12	02	81 05 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.2.19 0x1400/n, 0x1401/n, 0x1402/n RxPDO Communication Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1400 0x1401 0x1402	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	COB ID	Unsigned32	rw	No	See text
	2	Transmission type	Unsigned8	rw	No	See text
	3	Inhibit time	Unsigned16	rw	No	See text
	4	-	-	-	-	-
	5	Event time	Unsigned16	rw	No	See text

RxPDO Communication parameters:

0x1400/n RxPDO1 COB-ID Default value: 0x200 (=512) +Node ID
0x1401/n RxPDO2 COB-ID Default value: 0x300 (=768) +Node ID
0x1402/n RxPDO3 COB-ID Default value: 0x400 (=1024) +Node ID

These communication parameters define the COB-ID and transmission type used by the RxPDOs. Only sub-index 1,2 and 5 are used for RxPDOs. The default setting for the used COB-ID depends on the Node ID and can be changed. The default value for transmission type is 255 (event driven) and can also be changed (see table).

Object 0x1400/0x1401/0x1402 COB-ID				
Bit 31	Bit 30	Bit 29	Bit 11 ... 28	Bit 0 ... 10
valid	**0**	frame	**0**	11 bit CAN-ID

Bit 31: 0 = PDO existent/valid
1 = PDO non-existent/not valid

Bit 29: 0 = 11 Bit ID
1 = 29 Bit ID **NOT ALLOWED**

Bit 0 ... 10: 11 bit CAN-ID

RxPDO1 factory setting = valid
RxPDO2/3 factory setting = not valid

Object 0x1400/0x1401/0x1402 transmission type		
value	**meaning**	**description**
0	synchronous	Update RxPDO data on **each** SYNC
1 ... 240	synchronous	Update RxPDO data on **each** SYNC
~~*241 ... 251*~~	reserved	Value not allowed
~~*252*~~	synchronous/RTR	Value not allowed
~~*253*~~	asynchronous/RTR	Value not allowed
254	asynchronous	Event driven (manufacturer specific)
255	asynchronous	Event driven (profile specific) **default value**

Values 254 & 255 are handled identically. Update RxPDO data on each Rx.

Inhibit time:
The inhibit time for RxPDO is without function. Values can be entered, but are without further function.

Event time:
The event time is used as monitoring function for RxPDO's. If during the set time no RxPDO is received, one of the following faults is triggered:
202A Fault RxPDO1
202B Fault RxPDO2
202C Fault RxPDO3

Example*:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 14	02	00
Reply	581	4F	00 14	02	FF
Read Request	601	40	00 14	01	00
Reply	581	4F	00 14	01	01 02 00 00
Write Access	601	23	00 14	01	01 02 00 80
Reply *	581	60	00 14	01	00 00 00 00
Write Access	601	2F	00 14	02	05
Reply *	581	60	00 14	02	00
Write Access	601	23	00 14	01	01 02 00 00
Reply *	581	60	00 14	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

* Note, that Object 1400/1 Highest has to be deactivated first for the correct Write access for Object 1400/2.

12.2.20 0x1600/n, 0x1601/n, 0x1602 RxPDO Mapping Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1600 0x1601 0x1602	0	Number of mapped objects	Unsigned8	rw	No	2
	1	1st mapped obj.	Unsigned32	rw	No	See text
	2	2nd mapped obj.	Unsigned8	rw	No	See text
	3	3rd mapped obj.	Unsigned8	rw	No	See text
	4	4th mapped obj.	Unsigned8	rw	No	See text
	5	5th mapped obj.	Unsigned8	rw	No	See text
	6	6th mapped obj.	Unsigned8	rw	No	See text
	7	7th mapped obj.	Unsigned8	rw	No	See text
	8	8th mapped obj.	Unsigned8	rw	No	See text

RxPDO Mapping parameters:
0x1600/n RxPDO1
0x1601/n RxPDO2
0x1602/n RxPDO3

0x1600/0 = 0 = no objects mapped

0x1600/0 = 1 ... 8 = 1 ... 8 objects mapped

Mapping entry:

MSB			LSB
Object index		Subindex	Length (no. of bits)
High byte	Low byte	si	ll

Examples:

Mapping of 0x6040/0 *Control word* (unsigned16 = 10_{hex}) to 1st mapped object in RxPDO1:

0x1600/1 = 0x60400010

Mapping of 0x60C1/1 *interpolation data record 1* (integer32 = 20_{hex}) to 2nd mapped object in RxPDO1:

0x1600/2 = 0x60C10120

Refer to chapter 12.1 for a tabular overview of all objects and their corresponding data types.

Default mapping

RxPDO1	0x1600/0	0x1600/1	0x1600/2	0x1600/3...8
	2	0x6040 Control word	0x6042 target velocity	0x00000000
RxPDO2	0x1601/0			
	0	No mapping		
RxPDO3	0x1602/0			
	0	No mapping		

Example*:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 16	01	00 00 00 00
Reply	581	43	00 16	01	10 00 40 60
Write Access	601	2F	00 16	00	00
Reply *	581	60	00 16	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

* Note, that Object 1400/1 Highest bit has to be deactivated first for the correct Write access for Object 1600/n. See also the Mapping sequence described in the following.

Mapping Sequence

The mapping sequence requires five steps:

Step 1: Set PDO to “not valid” (0x1400, Subindex 1, Bit 31 = 1)

Step 2: Set sub index 0 to 0 (deactivate current mapping, 0x1600, Sub index 0 = 0)

Step 3: Set sub index 1 ... n to the new objects (0x1600, Subindex 1..n = new object)

Step 4: Set sub index 0 to the number of mapped objects (activate new mapping, 0x1600, Subindex 0 = n)

Step 5: Set PDO valid (0x1400, Subindex 1, Bit 31 = 0)

TxPDO 0x1600 is used exemplary above. The same procedure applies to 0x1601 and 0x1602. In these cases, 0x1400 has to be substituted accordingly with 0x1401 or 0x1402.

Example (Node ID = 1):

	COB ID	Control byte	Index LSB MSB	Subindex Subindex	Data LSB ...	Data ... MSB
Step 1:	601	23	00 14	01	01 02	00 80
Response	581	60	00 14	01	00 00	00 00
Step 2:	601	2F	00 16	00	00	
Response	581	60	00 16	00	00	
Step 3.1:	601	23	00 16	01	10 00	42 60
Response	581	60	00 16	01	00 00	00 00
Step 3.2	601	23	00 16	02	10 00	40 60
Response	581	60	00 16	02	00 00	00 00
Step 3.3	601	23	00 16	03	08 00	60 60
Response	581	60	00 16	03	00 00	00 00
Step 4:	601	2F	00 16	00	03	
Response	581	60	00 16	00	00	
Step 5:	601	23	00 14	01	01 02	00 00
Response	581	60	00 14	01	00 00	00 00

Resulting mapping

Target velocity (0x6042)	Control word (0x6040)	Modes of operation (0x6060)
00 00	00 00	00

This example shows the necessary telegrams with the according responses of the device.
The mapping is only stored in RAM and therefore are lost after a power restart. To store the mapping into EEPROM (power-fail safe) refer to chapter 12.2.11.

The number of mappable objects depends on the object's length.
The maximum number of bytes that can be mapped is 8.

12.2.21 0x1800/n, 0x1801/n, 0x1802/n TxPDO Communication Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1800 0x1801 0x1802	0	Highest sub-index supported	Unsigned8	ro	No	5
	1	COB ID	Unsigned32	rw	No	See text
	2	Transmission type	Unsigned8	rw	No	255
	3	Inhibit time	Unsigned16	rw	No	See text
	4	-	-	-	-	-
	5	Event time	Unsigned16	rw	No	See text

TxPDO Communication parameters:

0x1800/n TxPDO1 COB-ID Default value: 0x180 (=384) +Node ID
0x1801/n TxPDO2 COB-ID Default value: 0x280 (=640) +Node ID
0x1802/n TxPDO3 COB-ID Default value: 0x380 (=896) +Node ID

These communication parameters define the COB-ID and transmission type used by the TxPDOs. The default setting for the COB-ID depends on the Node ID and can be changed. The default value for the transmission type is 255 (event driven) and can also be changed (see table).

Object 0x1800/0x1801//1802 COB-ID				
Bit 31	Bit 30	Bit 29	Bit 11 ... 28	Bit 0 ... 10
valid	**0**	frame	**0**	11 bit CAN-ID

Bit 31: 0 = PDO existent / valid
1 = PDO non-existent / not valid

Bit 29: 0 = 11 Bit ID
1 = 29 Bit ID **NOT ALLOWED**

Bit 0 ... 10: 11 bit CAN-ID

TxPDO1 factory setting = valid
TxPDO2/3 factory setting = not valid

Object 0x1800/0x1801/0x1802 transmission type		
value	**meaning**	**description**
0	Synchronous	Update TxPDO data and send on SYNC **only** when data has changed
1 ... 240	Synchronous	Update TxPDO data and send on **each** "n" SYNC
~~241 ... 251~~	Reserved	Value not allowed
252	synchronous/RTR	Update TxPDO data on SYNC and send **on following** RTR
253	asynchronous/RTR	Update TxPDO data and send on RTR
254	asynchronous	Event driven (manufacturer specific)
255	asynchronous	Event driven (profile specific) **default value**

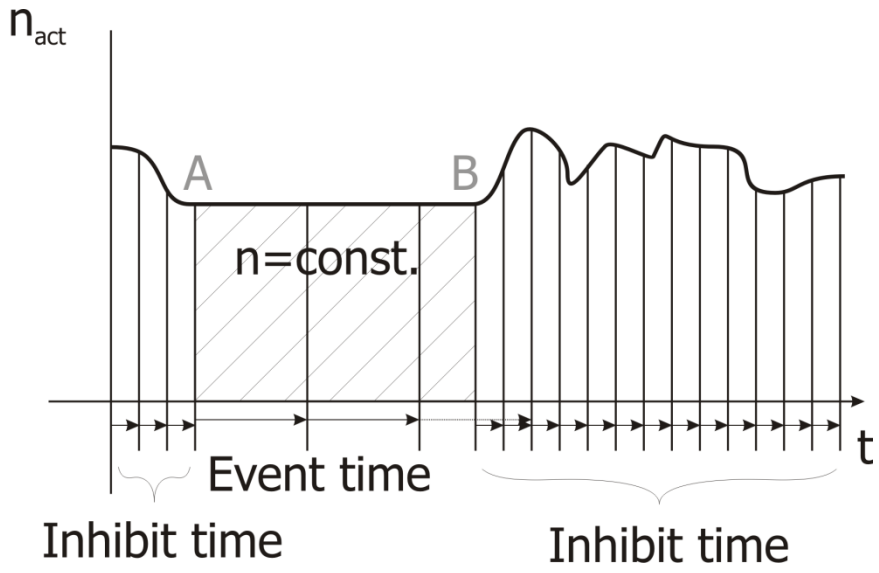
Values 254 + 255 are handled identically. Send TxPDO on data change or event time.

Inhibit time: The inhibit time is the minimum time distance between two consecutive TxPDOs for asynchronous TxPDOs. During the inhibit time, the TxPDO is not send again. Therefore a value change occurring in this time is send earliest after the inhibit time has elapsed. The value range is 0...65535.
The inhibit time is set in hundreds of microseconds, e. g. a value of 300 is 300 *100 us = 30 ms.

The device internal time resolution for the inhibit time is in milliseconds, the last digit is always converted to "0". An inhibit time value = 37 is truncated to 30 [3.7 ms → 3 ms].
Values less than 10 are interpreted as 0.

Event time: The event time is the time distance between two consecutive TxPDOs whenever the TxPDO data has not changed (cycle time). If the inhibit time is set to zero the TxPDO is only sent on a change of the TxPDO's data. The value range is 0...65535.
The event time is set in milliseconds, e.g. a value of 2000 = 2000 ms.

Example Event time & Inhibt time:
The actual speed value is transferred via TxPDO. The value is updated after the inhibit time has elapsed. At time A, the value remains constant. During this time, the value is updated after the Event time has elapsed. At time B, the value changes and is transmitted via TxPDO. The value changes again frequently and is only updated after the inhibit time has elapsed



Sub index 4:
Sub-index 4 is included for compatibility reasons. An SDO read/write access to sub index 4 results in an SDO abort.

Example*:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 18	02	00
Reply	581	4F	00 18	02	FF
Read Request	601	40	00 18	01	00
Reply	581	4F	00 18	01	81 01 00 00
Write Access	601	23	00 18	01	81 01 00 80
Reply *	581	60	00 18	01	00 00 00 00
Write Access	601	2F	00 18	02	05
Reply *	581	60	00 18	02	00
Write Access	601	23	00 18	01	81 01 00 00
Reply *	581	60	00 18	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

* Note, that Object 1800/1 highest bit has to be deactivated first for the correct Write access for Object 1800/2.

12.2.22 0x1A00/n, 0x1A01/n, 0x1A02/n TxPDO Mapping Parameters

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x1A00 0x1A01 0x1A02	0	Number of mapped objects	Unsigned8	rw	No	2
	1	1st mapped obj.	Unsigned32	rw	No	See text
	2	2nd mapped obj.	Unsigned32	rw	No	See text
	3	3rd mapped obj.	Unsigned32	rw	No	See text
	4	4th mapped obj.	Unsigned32	rw	No	See text
	5	5th mapped obj.	Unsigned32	rw	No	See text
	6	6th mapped obj.	Unsigned32	rw	No	See text
	7	7th mapped obj.	Unsigned32	rw	No	See text
	8	8th mapped obj.	Unsigned32	rw	No	See text

TxPDO Mapping parameters:
0x1A00/n TxPDO1
0x1A01/n TxPDO2
0x1A02/n TxPDO3

0x1A00/0 = 0 = no object mapped

0x1A00/0 = 1 ... 8 = 1 ... 8 objects mapped

Mapping entry:

MSB			LSB
Object index		Subindex	Length (no. of bits)
High byte	Low byte	si	ll

Examples:

Mapping of 0x6041/0 *Status word* (unsigned16) to 1st mapping object inTxPDO1:

0x1A00/1 = 0x60410010

Mapping of 0x6064/0 *position actual value* (integer32) to 2nd mapping object in TxPDO1:

0x1A00/2 = 0x60640020

Default mapping

TxPDO1	0x1A00/0	0x1A00/1	0x1A00/2	0x1A00/3...8
	2	0x6041 Status word	0x6044 control effort	0x00000000
TxPDO2	0x1A01/0			
	0	No mapping		
TxPDO3	0x1A02/0			
	0	No mapping		

The number of mappable objects depends on the object's length.
The maximum number of bytes that can be mapped is 8.

Example*:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 1A	01	00 00 00 00
Reply	581	43	00 1A	01	10 00 41 60
Write Access	601	2F	00 1A	00	00
Reply *	581	60	00 1A	00	00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

* Note, that Object 1400/1 Highest has to be deactivated first for the correct Write access for Object 1600/n. See also the Mapping sequence described in the following.

Mapping Sequence

The mapping sequence requires five steps:

Step 1: Set PDO to “not valid” (0x1800, subindex 1, Bit 31 = 1)

Step 2: Set subindex 0 to 0 (deactivate current mapping, 0x1A00, subindex 0 = 0)

Step 3: Set subindex 1 ... n to the new objects (0x1A00, subindex 1..n = new object)

Step 4: Set subindex 0 to the number of mapped objects (activate new mapping, 0x1A00, subindex 0 = n)

Step 5: Set PDO valid (0x1800, subindex 1, Bit 31 = 0)

TxPDO 0x1A00 is used exemplary above. The same procedure applies to 0x1A01 and 0x1A02. In these cases, 0x1800 has to be substituted accordingly with 0x1801 or 0x1802.

Example (Node ID = 1):

	COB ID	Control byte	Index	Sub index	Data	Data
			LSB MSB	Sub index	LSB ...	... MSB
Step 1:	601	23	00 18	01	84 01	00 80
Response	581	60	00 18	01	00 00	00 00
Step 2:	601	2F	00 1A	00	00 00	
Response	581	60	00 1A	00	00 00	
Step 3.1:	601	23	00 1A	01	10 00	44 60
Response	581	60	00 1A	01	00 00	00 00
Step 3.2	601	23	00 1A	02	10 00	41 60
Response	581	60	00 1A	02	00 00	00 00
Step 3.3	601	23	00 1A	03	10 00	01 30
Response	581	60	00 1A	03	00 00	00 00
Step 3.4	601	23	00 1A	04	10 00	02 30
Response	581	60	00 1A	04	00 00	00 00
Step 4:	601	2F	00 1A	00	04 00	
Response	581	60	00 1A	00	00 00	
Step 5:	601	23	00 18	01	84 01	00 00
Response	581	60	00 1A	00	00 00	00 00

Resulting mapping			
Control effort (0x6044)	Status word (0x6041)	Digital In actual values (0x3001)	Digital In actual values (0x3002)
00 00	00 00	00	00

12.3 Manufacturer objects (0x2nnn) – Parameter access

For direct write/read access to inverter parameters via the SDO channel, a parameter is addressed via index and sub-index. Index and sub-index are used as follows for accessing the inverter parameters:

Index = Parameter number + 0x2000
Sub-index = required data set (0, 1 ... 4, 5, 6 ... 9)

The mapping of numeric data is always in integer or long data type. Values which contain decimal places are extended accordingly:
(e.g. value 17.35 is transmitted as 1735)

12.3.1 Handling of data sets/cyclic writing of the parameters

Access to the parameter values is carried out on the basis of the parameter number and the required data set. There are parameters which only have one value (data set 0), as well as parameters which have four values (data sets 1...4). The latter are used for the data set change-over of a parameter.

If parameters with four data values are set via data set = 0, all four data sets are set to the same transmitted value. A read access with data set = 0 to such parameters is only successful if all four data sets are set to the same value. If this is not the case an error is reported.

NOTE

The values are entered automatically into the EEPROM on the controller. If values are to be written cyclically, there must be no entry into the EEPROM, as it only has a limited number of admissible writing cycles (about 1 million cycles). If the number of admissible writing cycles is exceeded, the EEPROM is destroyed.

To avoid this, cyclically written data can be entered exclusively into the RAM without a writing cycle taking place on the EEPROM. The data are volatile, i.e., they are lost on power-off and have to be written again after power-on.
This mechanism is activated by the target data set being increased by five in the specification of the data set.

Writing to a virtual data set in the RAM

Parameter	EEPROM	RAM
Data set 0	0	5
Data set 1	1	6
Data set 2	2	7
Data set 3	3	8
Data set 4	4	9

12.3.1.1 SDO examples (expedited transfer only)

Writing parameters:

Client ➔ Server SDO Download (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
		LSB	MSB	0xnn				
	0x2B	**uint/int**			LSB	MSB	--	--
	0x23	**long**			LSB	...	...	MSB

Server ➔ Client Download Response ➔ writing process free of errors

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
	0x60	LSB	MSB	0xnn	-			

Server ➔ Client Abort SDO Transfer ➔ writing process with error

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
	0x80	LSB	MSB	0xnn	Error code			

If an error occurs during the writing process, the corresponding error code is given in Bytes 4 ... 7.

Reading parameters:

Client ➔ Server SDO Upload (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
	0x40	LSB	MSB	0xnn	-			

Server ➔ Client Upload Response ➔ reading process without errors

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
		LSB	MSB	0xnn	LSB			MSB
	0x4B	**uint/int**			LSB	MSB	--	--
	0x43	**long**			LSB	...	...	MSB

Server ➔ Client Abort SDO Transfer ➔ reading process faulty

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
	0x80	LSB	MSB	0xnn	Error code			

If an error occurs during the reading process, the corresponding error code is given in Bytes 4 ... 7. Check chapter11.3.3 for the error code list for SDO abort.

12.3.1.2 Examples of writing parameters

Example 1 Write parameter *Rated Speed* **372** (type: uint) in data set 2 with the parameter value 2980.
Index = 372 + 0x2000 = 0x2174, value = 2980 = 0x0BA4

Client ➔ Server SDO Download (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x2B**	0x74	0x21	0x02	0xA4	0x0B	--	--

Example 2 Writing parameter *Warning Limit Heat Sink Temp.* **407** (type: int) in data set 0 with the parameter value -15.
Index = 407 + 0x2000 = 0x2197, value = -15 = 0xFFF1

Client ➔ Server SDO Download (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x2B**	0x97	0x21	0x00	0xF1	0xFF	--	--

Example 3 Writing parameter *Fixed frequency 1* **480** (type: long) in data set 1 with the parameter value 100.00 Hz.
Index = 480 + 0x2000 = 0x21E0, value = 10000 = 0x00002710

Client ➔ Server SDO Download (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x23**	0xE0	0x21	0x01	0x10	0x27	0x00	0x00

Example 4 Writing parameter *Fixed Frequency 1* **480** (type: long) in data set 3 with the parameter value -50.00 Hz.
Index = 480 + 0x2000 = 0x21E0, value = -5000 = 0xFFFFEC78

Client ➔ Server SDO Download (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x23**	0xE0	0x21	0x03	0x78	0xEC	0xFF	0xFF

If an error occurs during the writing process, the corresponding error code is given in Bytes 4 ... 7. Check chapter 11.3.3 for the error code list for SDO abort.

Using Write accesses for parameters (objects 0x2nnn = index), the sub-index is used to define the Write access into EEPROM or RAM. Please refer to chapter 12.3.1 “Handling of data sets/cyclic writing”.

12.3.1.3 Examples of reading parameters

Example 1 Read parameter *Rated speed* **372** (type: uint) in data set 2 with the current parameter value 1460.
Index = 372 + 0x2000 = 0x2174, value = 1460 = 0x05B4

Client ➔ Server SDO Upload (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x40**	0x74	0x21	0x02	--	--	--	--

Server ➔ Client Upload Response

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x581	**0x4B**	0x74	0x21	0x02	0xB4	0x05	--	--

Example 2 Read parameter *Warning Limit Heat Sink Temp.* **407** (type: int) in data set 0 with the current parameter value -5.
Index = 407 + 0x2000 = 0x2197, value = -5 = 0xFFFB

Client ➔ Server SDO Upload (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x40**	0x97	0x21	0x00	--	--	--	--

Server ➔ Client Upload Response

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x581	**0x4B**	0x97	0x21	0x00	0xFB	0xFF	--	--

Example 3 Read parameter *Fixed Frequency 1* **480** (type: long) in data set 1 with the current parameter value 75.00 Hz.
Index = 480 + 0x2000 = 0x21E0, value = 7500 = 0x00001D4C

Client ➔ Server SDO Upload (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x40**	0xE0	0x21	0x01	--	--	--	--

Server ➔ Client Upload Response

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x581	**0x43**	0xE0	0x21	0x01	0x4C	0x1D	0x00	0x00

Example 4 Reading parameter *Fixed Frequency 1* **480** (type: long) in data set 3 with the current parameter value -10.00 Hz.
Index = 480 + 0x2000 = 0x21E0, value = -1000 = 0xFFFFFC18

Client ➔ Server SDO Upload (expedited)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x40**	0xE0	0x21	0x03	--	--	--	--

Server ➔ Client Upload Response

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x581	**0x43**	0xE0	0x21	0x03	0x18	0xFC	0xFF	0xFF

If an error occurs during the reading process, the corresponding error code is given in Bytes 4 ... 7. Check chapter 11.3.3 for the error code list for SDO abort.

12.3.1.4 Example to Write parameters via Segmented Transfer

Write Parameter *User Name* **029** (Type String), in data set 0 with the parameter value “Bonfiglioli Vectron CANopen” (= 27 characters = 0x1B characters).
Index = 29 + 0x2000 = 0x201D

ASCII	B	o	n	f	i	g	l
Hexadecimal	0x42	0x6F	0x6E	0x66	0x69	0x67	0x6C
ASCII	i	o	l	i	(blank)	V	e
Hexadecimal	0x69	0x6F	0x6C	0x69	0x20	0x56	0x65
ASCII	c	t	r	o	n	(blank)	C
Hexadecimal	0x63	0x74	0x72	0x6F	0x6E	0x20	0x43
ASCII	A	N	o	p	e	n	
Hexadecimal	0x41	0x4E	0x6F	0x70	0x65	0x6E	

Initiate Sequence Client ➔ Server SDO Initiate Download Request (segmented)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Sub-index	Data			
0x601	**0x21**	0x1D	0x20	0x00	1B	00	00	00

Server ➔ Client Initiate Download Response (response: 9 bytes to be send)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Sub-index	Data			
0x581	**0x60**	0x1D	0x20	0x00	0x00	0x00	0x00	0x00

1st Segment Server ➔ Client Upload Segment Request (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x00**	0x42	0x6F	0x6E	0x66	0x69	0x67	0x6C

Server ➔ Client Upload Segment Response (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x20**	0x00	0x00	0x00	0x09	0x00	0x00	0x00

2nd Segment Server ➔ Client Upload Segment Request (bytes 8...14)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x10**	0x69	0x6F	0x6C	0x69	0x20	0x56	0x65

Server ➔ Client Upload Segment Response (bytes 8...14)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x30**	0x00	0x00	0x00	0x00	0x00	0x00	0x00

3rd Segment Server ➔ Client Upload Segment Request (bytes 15...21)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x00**	0x63	0x74	0x72	0x6F	0x6E	0x20	0x43

Server ➔ Client Upload Segment Response (bytes 15...21)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x20**	0x00	0x00	0x00	0x00	0x00	0x00	0x00

4th Segment Server ➔ Client Upload Segment Request (bytes 22...28)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x13**	0x41	0x4E	0x6F	0x70	0x65	0x6E	0x00

Server ➔ Client Upload Segment Response (bytes 22...28)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x30**	0x00	0x00	0x00	0x00	0x00	0x00	0x00

12.3.1.5 Examples to Read parameters via Segmented Transfer

Example 1 Read Parameter *Inverter Software Version* **012** (Type String), in data set 0 with the actual parameter value "5.2.0 STO".
Index = 12 + 0x2000 = 0x200C, Value = "5.2.0 STO"

ASCII	5	.	2	.	0	(blank)	S	T	O
Hexadec-imal	0x35	0x2E	0x32	0x2E	0x30	0x20	0x53	0x54	0x4F

Initiate Sequence Client ➔ Server SDO Initiate Download Request (segmented)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x601	**0x40**	0x0C	0x20	0x00	--	--	--	--

Server ➔ Client Initiate Download Response (response: 9 bytes to be send)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Subindex	Data			
0x581	**0x41**	0x0C	0x21	0x00	**0x09**	0x00	0x00	0x00

1st Segment Server ➔ Client Upload Segment Request (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x60**	0x00	0x00	0x00	0x09	0x00	0x00	0x00

Server ➔ Client Upload Segment Response (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x00**	0x35	0x2E	0x32	0x2E	0x30	0x20	0x53

2nd Segment Server ➔ Client Upload Segment Request (bytes 8...9)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x70**	0x00	0x00	0x00	0x09	0x00	0x00	0x00

Server ➔ Client Upload Segment Response (bytes 8...9)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x1B**	0x54	0x4F	0x00	0x09	0x00	0x00	0x00

Example 2 Reading of Parameter *User name* **029** (Type String), in data set 0 with the actual parameter value “CANopen device”.
Index = 29 + 0x2000 = 0x201D, Wert = “CANopen device”

ASCII	C	A	N	o	p	e	n
Hexadec-imal	0x43	0x41	0x4E	0x6F	0x70	0x65	6E
ASCII	(blank)	d	e	v	i	c	e
Hexadec-imal	0x20	0x64	0x65	0x76	0x69	0x63	0x65

Initiate Sequence Client ➔ Server SDO Initiate Download Request (segmented)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Sub-index	Data			
0x601	**0x40**	0x0C	0x20	0x00	0x00	0x00	0x00	0x00

Server ➔ Client Initiate Download Response (response: 9 bytes to be send)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Index		Sub-index	Data			
0x581	**0x41**	0x0C	0x21	0x00	**0x09**	0x00	0x00	0x00

1st Segment Server ➔ Client Upload Segment Request (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x60**	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Server ➔ Client Upload Segment Response (bytes 1...7)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x00**	0x43	0x41	0x4E	0x6F	0x70	0x65	6E

2nd Segment Server ➔ Client Upload Segment Request (bytes 8...14)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x601	**0x70**	0x00	0x00	0x00	0x09	0x00	0x00	0x00

Server ➔ Client Upload Segment Response (bytes 8...14)

	0	1	2	3	4	5	6	7
COB ID	Control byte	Data						
0x581	**0x11**	0x20	0x64	0x65	0x76	0x69	0x63	0x65

12.3.2 Handling of index parameters/cyclic writing

Index Parameters are used for different ACU functionalities.
Instead of the 4 data sets 16 or 32 indexes are used with these parameters. The addressing of the individual index is done via an index access parameter separated by the functionality. The selection to write into EEPROM or RAM is done via the index access parameter.

Function	Parameter	Index range		Index access parameter
		Write EEPROM and Read	Write RAM	
Positioning	**1202** *Target Position / Distance* **1203** *Speed* **1204** *Acceleration* **1205** *Ramp Rise time* **1206** *Deceleration* **1207** *Ramp Fall time* **1208** *Motion mode* **1209** *Touch-Probe-Window* **1210** *Touch-Probe-Error:Next Motion Block* **1211** *No. of Repetitions* **1212** *Delay* **1213** *Delay: Next Motion Block* **1214** *Event 1* **1215** *Event1: Next Motion Block* **1216** *Event 2* **1217** *Event2: Next Motion Block* **1218** *Digital Signal 1* **1219** *Digital Signal 2* **1247** *Digital Signal 1* **1248** *Digital Signal 2* **1260** *Interrupt-Event 1* **1261** *Int.-Event 1: Eval.-Mode* **1262** *Int.-Event 1: Next Motion Block* **1263** *Interrupt- Event 2* **1264** *Int.-Event 2: Eval.-Mode* **1265** *Int.-Event 2: Next Motion Block*	0[1)]; 1...32	33[1)]; 34...65	**1200** Write **1201** Read
PLC Function (Function table)	**1343** *FT-instruction* **1344** *FT-input 1* **1345** *FT-input 2* **1346** *FT-input 3* **1347** *FT-input 4* **1348** *FT-Parameter 1* **1349** *FT-Parameter 2* **1350** *FT-target output 1* **1351** *FT- target output 2* **1352** *FT-commentary*	0[1)]; 1...32	33[1)]; 34...65	**1341** Write **1342** Read
Multiplexer	**1252** *Mux Input*	0[1)]; 1...16	17[1)]; 18...33	**1250** Write **1251** Read
CANopen Multiplexer	**1422** *CANopen Mux Input*	0[1)]; 1...16	17[1)]; 18...33	**1420** Write **1421** Read

1) If the index access parameter is set = 0, all indexes are accessed to write into EEPROM. Selection 17 and 33 respectively, write all indexes into RAM.

NOTE

The values are entered automatically into the EEPROM on the controller. If values are to be written cyclically, there must be no entry into the EEPROM, as it only has a limited number of admissible writing cycles (about 1 million cycles). If the number of admissible writing cycles is exceeded, the EEPROM is destroyed.

To avoid this, cyclically written data can be entered exclusively into the RAM without a writing cycle taking place on the EEPROM. The data are volatile, i.e., they are lost on power-off and have to be written again after power-on.

12.3.2.1 Example Writing an index parameter

Typically an index parameter is written during commissioning or regularly at simple positioning applications.

Writing Parameter **1202** *Target Position / Distance* (Type long), in Index 1 into RAM (→index 34 for write access) with Parameter value 30000.
Index = 1200 + 0x2000 = 0x24B0, Value (int) = 34 = 0x0022
Index = 1202 + 0x2000 = 0x24B2, Value (long) = 30000 = 0x0000 7530

	COB ID	CB	Index	Sub-index	Data
Write Request P. 1200 to Index 34	0x601	0x2B	0xB0 0x24	0x05	0x22 0x00
Response	0x581	0x60	0xB0 0x24	0x05	0x00 0x00
Write Request P. 1202 to 30000 u	0x601	0x23	0xB2 0x24	0x00	0x30 0x75 0x00 0x00
Response	0x581	0x60	0xB2 0x24	0x00	0x00 0x00 0x00 0x00

If several parameter of an index should be changed, it is sufficient to set the index access parameter **1200** once at the beginning.

12.3.2.2 Example Reading an index parameter

To read an index parameter, first the index access parameter has to be set to the corresponding index. After that, the parameter can be read out.

Reading Parameter **1202** *Target Position / Distance* (Type long), in Index 1 with Parameter value 123000.
Index = 1201 + 0x2000 = 0x24B1, Value (int) = 1 = 0x0001
Index = 1202 + 0x2000 = 0x24B2, Value (long) = 123000 = 0x0001 E078

	COB ID	CB	Index	Sub-index	Data
Write Request P. 1201 to Index 1	0x601	0x2B	0xB1 0x24	0x05	0x01 0x00
Response	0x581	0x60	0xB1 0x24	0x05	0x00 0x00
Read Request P. 1202	0x601	0x40	0xB2 0x24	0x00	0x00 0x00 0x00 0x00
Response	0x581	0x43	0xB2 0x24	0x00	0x78 0xE0 0x01 0x00

If several parameter of an index should be read, it is sufficient to set the index access parameter **1201** once at the beginning.

12.4 Manufacturer objects (0x3000 ... 0x5FFF)

In addition to the device profile objects the following manufacturer specific objects are implemented.

12.4.1 0x3000/0 SYNC Jitter

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3000	0	SYNC Jitter	Unsigned16	rw	No	See Text

DS301 does not include an object for monitoring the jittering of the SYNC message. ACTIVE CUBE inverters monitor SYNC message jittering with object 0x3000/0 *SYNC Jitter* (given in multiples of micro seconds).

If the SYNC message is received outside the time defined by:

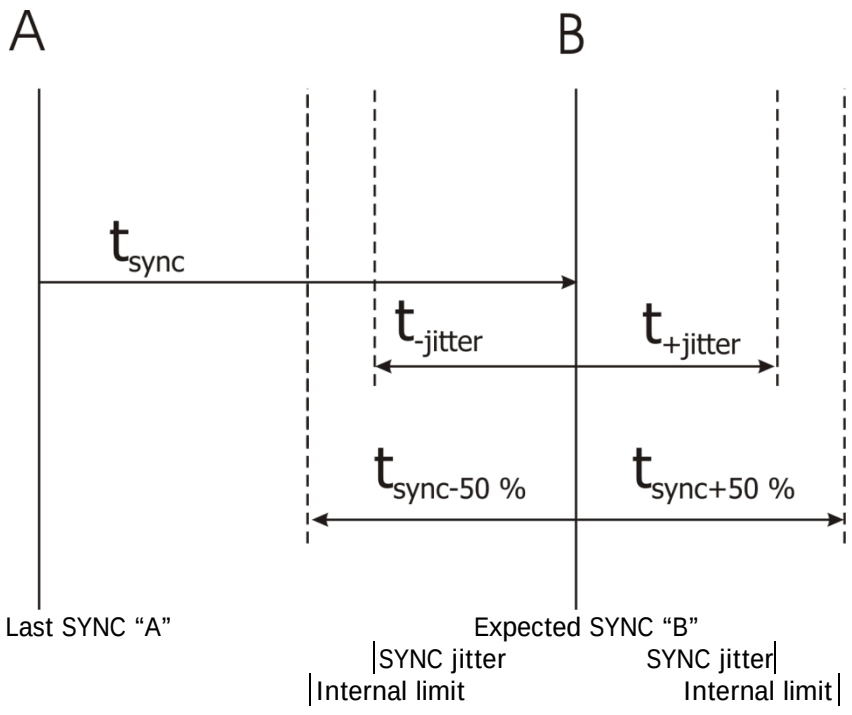
0x1006/0 ***communication cycle period*** **+/- 0x3000/0** ***SYNC Jitter***

a communication error event is triggered.

The value for object 0x3000/0 *SYNC Jitter* depends on the CANopen® master's capability for time accuracy. The value range is 0 ... 17.000 (µs) and is in addition internally restricted to 50% of the *communication cycle period* (object 0x1006/0 or the measured value).

If object 0x3000/0 *SYNC Jitter* is set to 0, there is no monitoring of the SYNC message time.

If object 0x3000/0 *SYNC Jitter* is set to ≠ 0 then monitoring of the SYNC message time is active. The jitter monitoring is independent of how the communication cycle period is determined (either set with object 0x1006/0 or by measuring).



Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 30	00	00 00
Reply	581	4B	00 30	00	00 00
Write Access	601	2B	00 30	00	10 00
Reply	581	60	00 30	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

Object 0x3000 *SYNC Jitter* is located in the application object area and is saved by objects 0x1010/3 *save application objects* and 0x1010/1 *save all objects*.

12.4.2 0x3001/0 Digital In actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3001	0	Digital In actual value	Unsigned16	ro	Tx	

Object 0x3001 *Digital In actual value* displays the current state of the digital inputs and of the multifunctional input 1 (if in *Operation mode* **452**-digital input) as described in parameter *Digital Inputs* **250**.

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	01 30	00	00 00
Reply	581	4B	01 30	00	06 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.3 0x3002/0 Digital Out actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3002	0	Digital Out actual value	Unsigned16	ro	Tx	

Object 0x3002 *Digital Out actual value* displays the current state of the up to 4 - depending on optional hardware - digital outputs and of the multifunctional output 1 (if in *Operation mode* **550** – digital) as described in parameter *Digital Outputs* **254**.

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	02 30	00	00 00
Reply	581	4B	02 30	00	03 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.4 0x3003/0 Digital Out set values

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3003	0	Digital Out set values	Unsigned8	rw	Rx	0

Via object 0x3003 there are 5 digital sources available for use with parameters, which require digital values.

Object 0x3003			
Bit	Source no.	Source name	Operation mode Digital output
0	810	Obj 0x3003 Digout 1	90/190
1	811	Obj 0x3003 Digout 2	91/191
2	812	Obj 0x3003 Digout 3	92/192
3	813	Obj 0x3003 Digout 4	93/193
4	814	Obj 0x3003 Digout 5	94/194

The value of object 0x3003 is limited from 0 to 31.

No.	Object	Min.	Max.
0x3003/0	Digital Out set values	0	31 (= 0x1F)

Digital outputs use these sources as operation mode 90 ... 94 *Obj 0x3003 DigOut 1 ... 5* respectively inverted as 190 ... 194 *inv. Obj 0x3003 DigOut 1 ... 5* (see e.g. parameter *Op. Mode Digital Output 1* **530***).* The mapping of this object bits to the outputs is arbitrary.

Example:

Function	Parameter no.	Choice list (excerpt)
Op. Mode Digital Output 3	532	0 - OFF 1 - Ready or Standby Signal 2 - Run Signal ... 43 - External Fan 90 - Obj 0x3003 Digout 1 91 - Obj 0x3003 Digout 2 92 - Obj 0x3003 Digout 3 93 - Obj 0x3003 Digout 4 94 - Obj 0x3003 Digout 5 ... 143 - inv. External Fan 190 - inv. Obj 0x3003 Digout 1 191 - inv. Obj 0x3003 Digout 2 192 - inv. Obj 0x3003 Digout 3 193 - inv. Obj 0x3003 Digout 4 194 - inv. Obj 0x3003 Digout 5 ...

The sources can be chosen directly by the selection of 810...814 *Obj 0x3003 DigOut 1 ... 5* from the parameters choice list. This can be used e.g. for direct setting of Boolean inputs.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	03 30	00	00
Reply	581	4F	03 30	00	03
Write Access	601	2F	03 30	00	10
Reply	581	60	03 30	00	00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

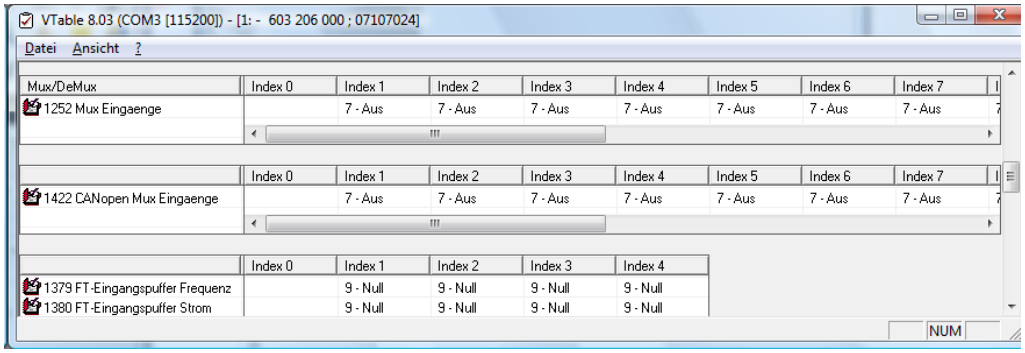
12.4.5 0x3004/0 Boolean Mux

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3004	0	Boolean Mux	Unsigned16	ro	Tx	

Via object 0x3004 up to 16 Boolean values can be transferred from the ACU to a PLC in a compressed manner. Each bit in 16 bit object 0x3004 displays the actual value of the connected Boolean source.

Bit number 0 ... 15 corresponds with index number 1 ... 16!

The sources for the 16 bits can be chosen from a choice list via the index parameter *CANopen® Mux Input*. **1422**. Parameters **1420** and **1421** are the associated write and read parameters which you have to set prior to writing/reading parameter **1422**. By using VTable this process is easier and more clearly laid out.
For writing and reading index parameters refer to chapter12.3.2 “Handling of index parameters/cyclic writing”.



Default value is 7 – Off.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	04 30	00	00 00
Reply	581	4B	04 30	00	03 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.6 0x3005/0 Boolean DeMux

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3005	0	Boolean DeMux	Unsigned16	rw	Rx	0

Via object 0x3005 up to 16 Boolean values can be written in a compressed manner. These values are available as sources which can be chosen by the selection of 832 ... 847 *Obj 0x3005 Demux Out 1...16* from a parameters choice list.

Obj. 0x3005		
Bit no.	**Source No.**	**Source name**
0	832	Obj. 0x3005 Demux Out 1
1	833	Obj. 0x3005 Demux Out 2
2	834	Obj. 0x3005 Demux Out 3
3	835	Obj. 0x3005 Demux Out 4
4	836	Obj. 0x3005 Demux Out 5
5	837	Obj. 0x3005 Demux Out 6
6	838	Obj. 0x3005 Demux Out 7
7	839	Obj. 0x3005 Demux Out 8
8	840	Obj. 0x3005 Demux Out 9
9	841	Obj. 0x3005 Demux Out 10
10	842	Obj. 0x3005 Demux Out 11
11	843	Obj. 0x3005 Demux Out 12
12	844	Obj. 0x3005 Demux Out 13
13	845	Obj. 0x3005 Demux Out 14
14	846	Obj. 0x3005 Demux Out 15
15	847	Obj. 0x3005 Demux Out 16

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	05 30	00	00 00
Reply	581	4B	05 30	00	05 00
Write Access	601	2B	05 30	00	20 00
Reply	581	60	05 30	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

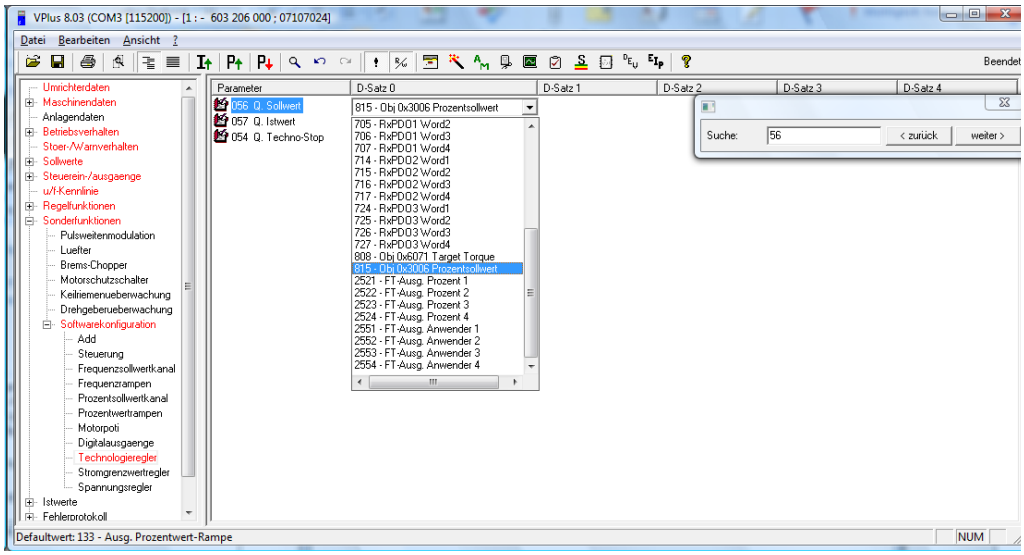
12.4.7 0x3006/0 Percentage set value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3006	0	Percentage set value	Unsigned16	rw	Rx	0

Via object 0x3006 it is possible to write to a percentage source like parameter *S. Reference Value* **056**.
The value of object 0x3006 is available as source which can be chosen by the selection of “815 - Obj 0x3006 Reference Percentage” from a parameters choice list.
The value of object 0x3006 is limited to -30000 to 30000 (corresponds to percentage values -300.00 %...300.00 %).

No.	Object	Min.	Max.
0x3006/0	Percentage set value	-30000 (= 0x8AD0)	30000 (= 0x7530)

Example: Technology controller parameter *S. Reference Value* **056**.



The percentage value is scaled as percent * 100 (e.g. 5678 represents 56.78%).

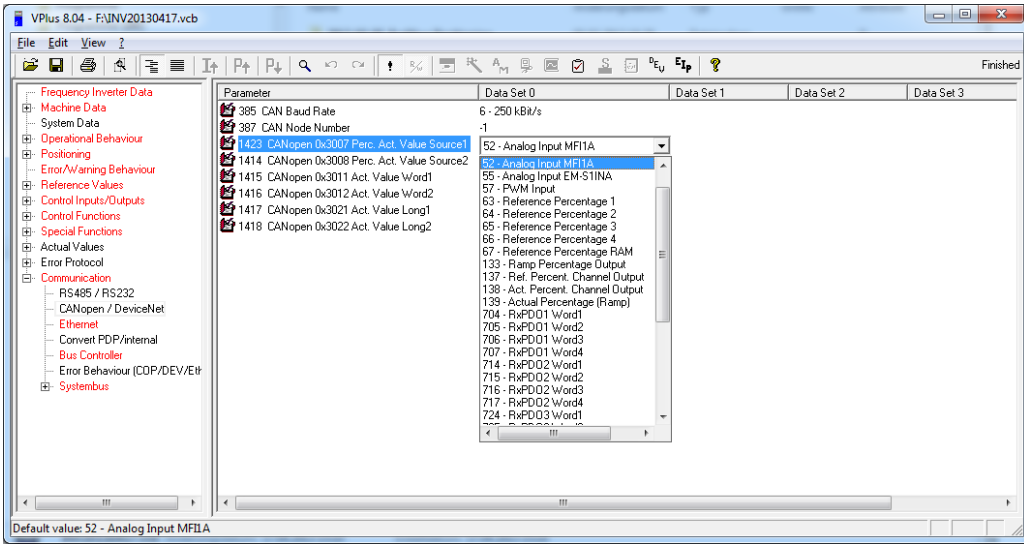
Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	06 30	00	00 00
Reply	581	4B	06 30	00	05 00
Write Access	601	2B	06 30	00	20 00
Reply	581	60	06 30	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.8 0x3007/0 Percentage actual value 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3007	0	Percentage actual value 1	Unsigned16	ro	Tx	

Object 0x3007 displays the value of a percentage source which is selectable via parameter *CANopen® Percentage Actual Value Source* **1423**.
Default source is 52 – Analog Input MFI1A.



The percentage value is scaled as percent * 100 (e.g. 5678 represents 56.78%).

Example:

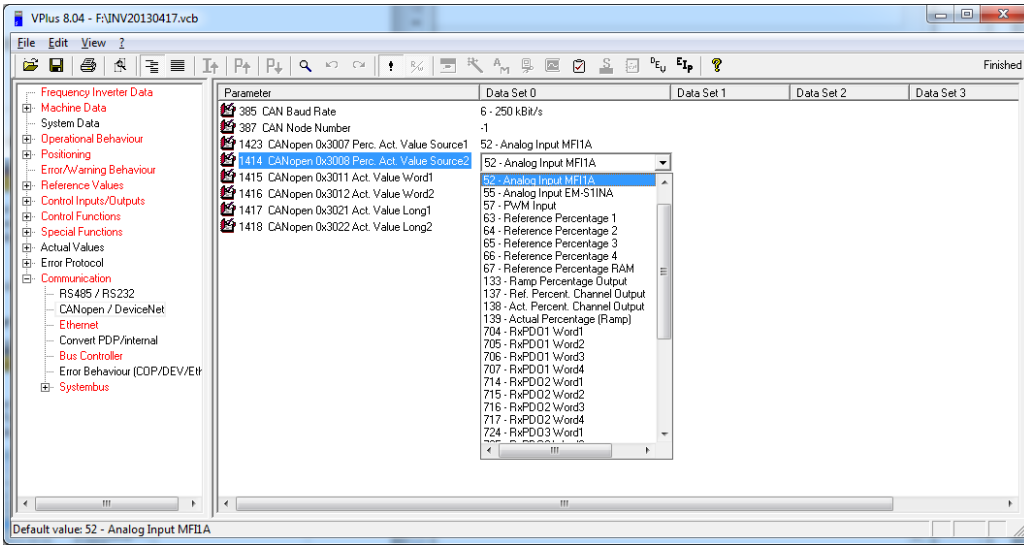
	COB ID	CB	Index	SI	Data
Read Request	601	40	07 30	00	00 00
Reply	581	4B	07 30	00	8F 13

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.9 0x3008/0 Percentage actual value 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3008	0	Percentage actual value 2	Unsigned16	ro	Tx	

Object 0x3008 displays the value of a percentage source which is selectable via parameter *CANopen® Percentage Actual Value Source 2* **1414**.
Default source is 52 – Analog Input MFI1A.



The percentage value is scaled as percent * 100 (e.g. 5678 represents 56.78%).

Example:

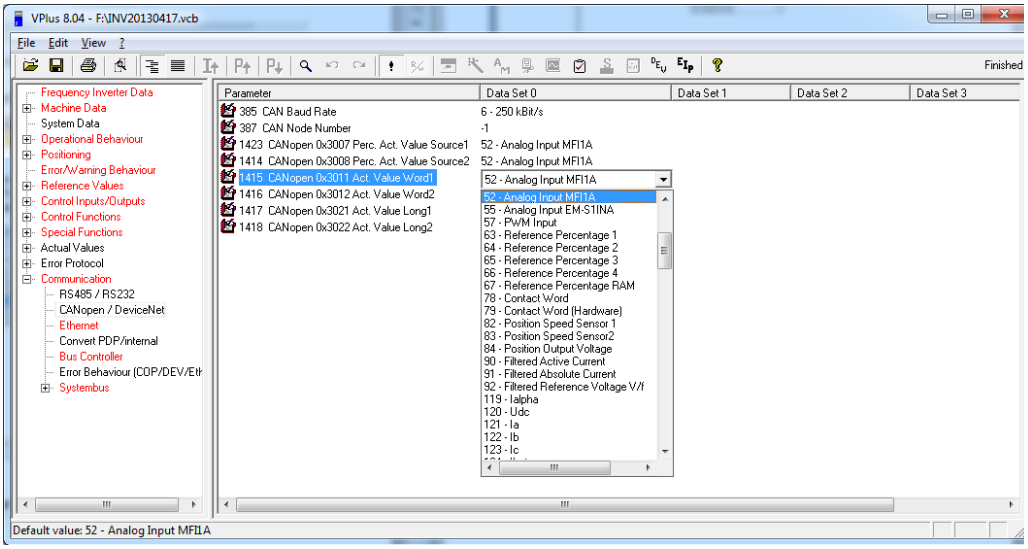
	COB ID	CB	Index	SI	Data
Read Request	601	40	08 30	00	00 00
Reply	581	4B	08 30	00	8F 13

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.10 0x3011/0 Actual value Word 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3011	0	Actual value Word 1	Unsigned16	ro	Tx	

Object 0x3011 displays the value of a word source which is selectable via parameter *CANopen Actual. Value Word 1* **1415**.
Default source is 52 – Analog Input MFI1A.



Example:

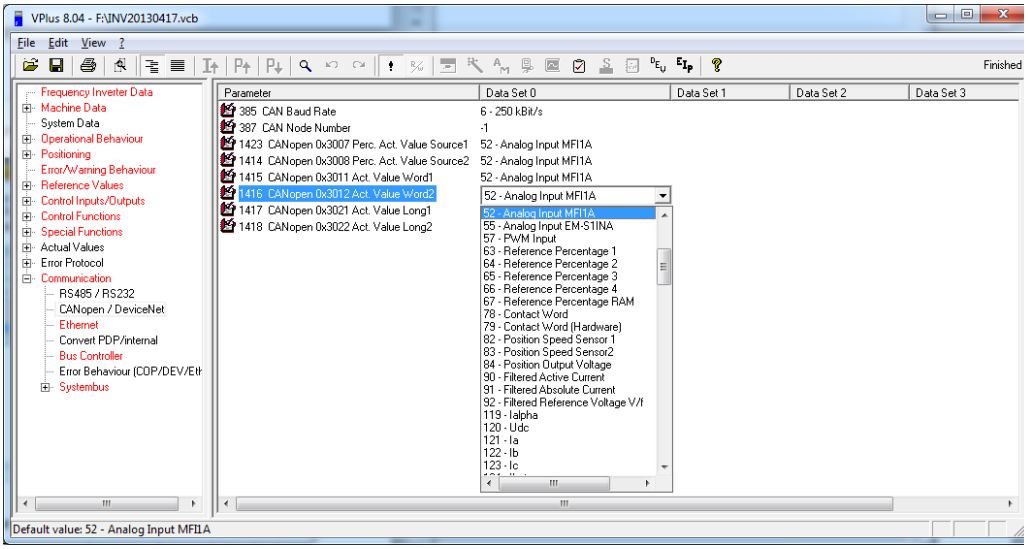
	COB ID	CB	Index	SI	Data
Read Request	601	40	11 30	00	00 00
Reply	581	4B	11 30	00	8F 13

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.11 0x3012/0 Actual value Word 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3012	0	Actual value Word 2	Unsigned16	ro	Tx	

Object 0x3012 displays the value of a word source which is selectable via parameter *CANopen® Actual Value Word 2* **1416**.
Default source is 52 – Analog Input MFI1A.



Example:

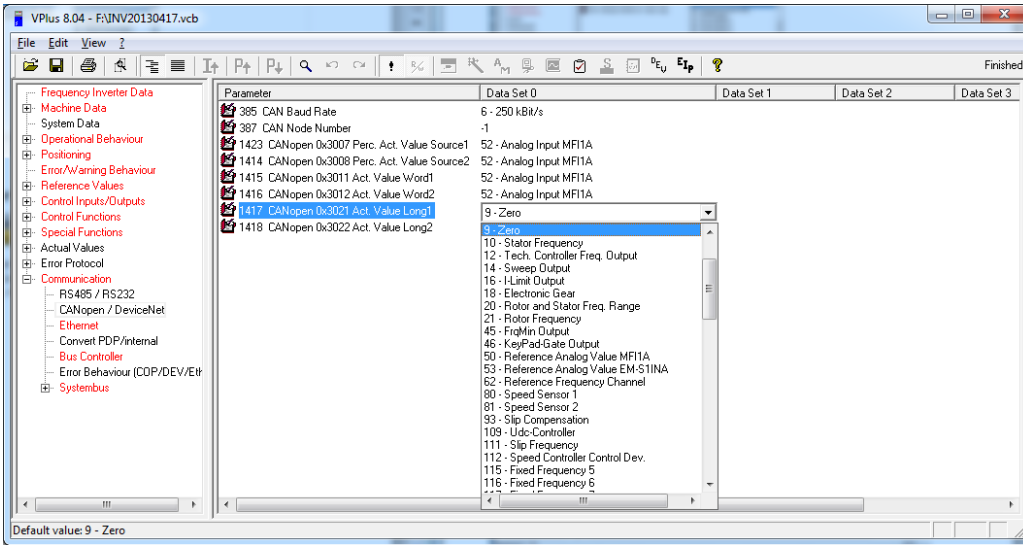
	COB ID	CB	Index	SI	Data
Read Request	601	40	12 30	00	00 00
Reply	581	4B	12 30	00	8F 13

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.12 0x3021/0 Actual value Long 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3021	0	Actual value Long 1	Unsigned32	ro	Tx	

Object 0x3021 displays the value of a Long source which is selectable via parameter *CANopen® Actual Value Long 1* **1417**.
Default source is 9 – Zero.



Example:

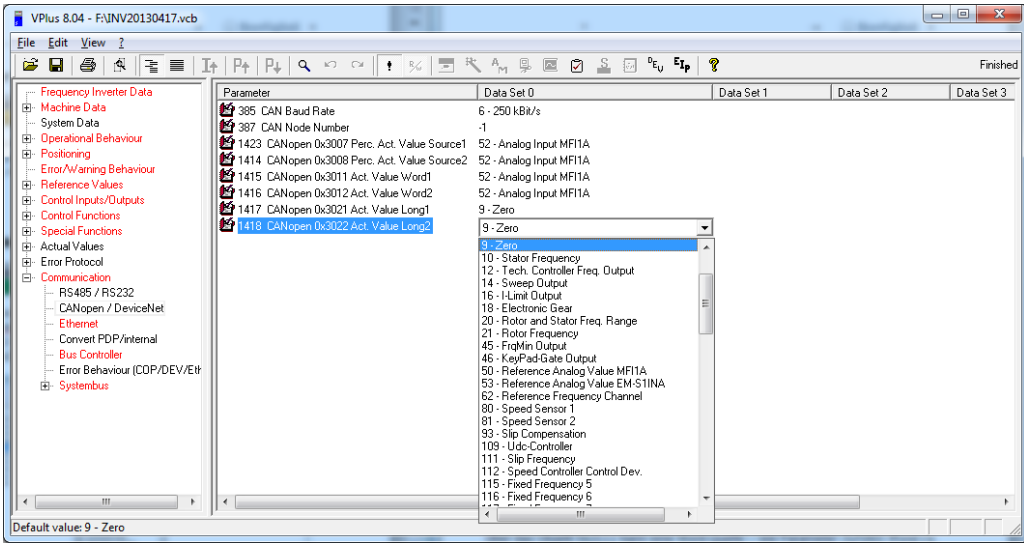
	COB ID	CB	Index	SI	Data
Read Request	601	40	21 30	00	00 00 00 00
Reply	581	43	21 30	00	8F 13 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.13 0x3022/0 Actual value Long 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3022	0	Actual value Long 2	Unsigned32	ro	Tx	

Object 0x3022 displays the value of a Long source which is selectable via parameter *CANopen® Actual Value Long 2* **1418**.
Default source is 9 – Zero.



Example:

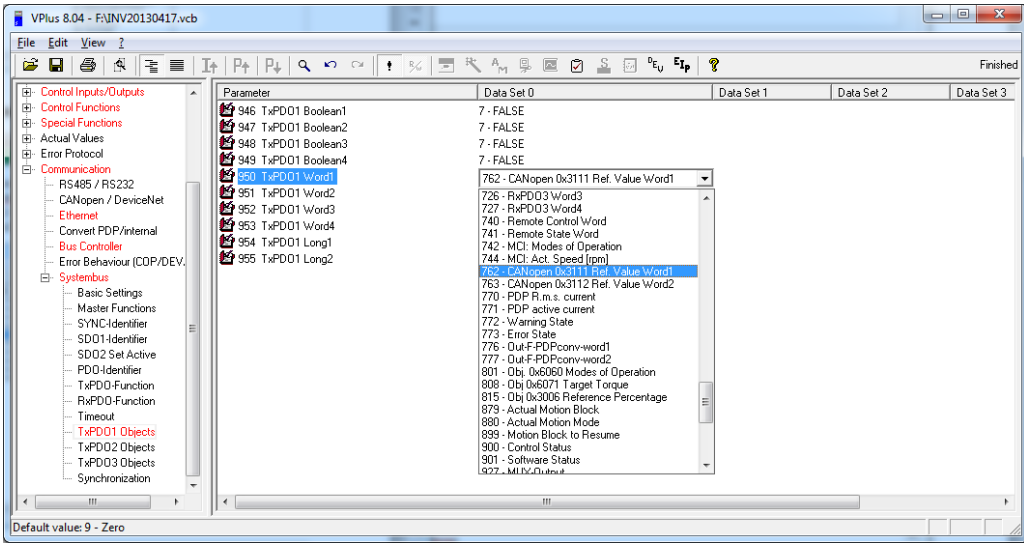
	COB ID	CB	Index	SI	Data
Read Request	601	40	22 30	00	00 00 00 00
Reply	581	4B	22 30	00	8F 13 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.14 0x3111/0 Ref. Value word 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3111	0	Ref. Value word 1	Unsigned16	rw	Rx	0

Via object 0x3111 it is possible to write to a Word source like parameter *TxPDO1 Word 1* **950** of the Systembus.
The value of object 0x3111 is available as source which can be chosen by the selection of “762 - CANopen 0x3111 Ref. Value” from a parameters choice list.



Example:

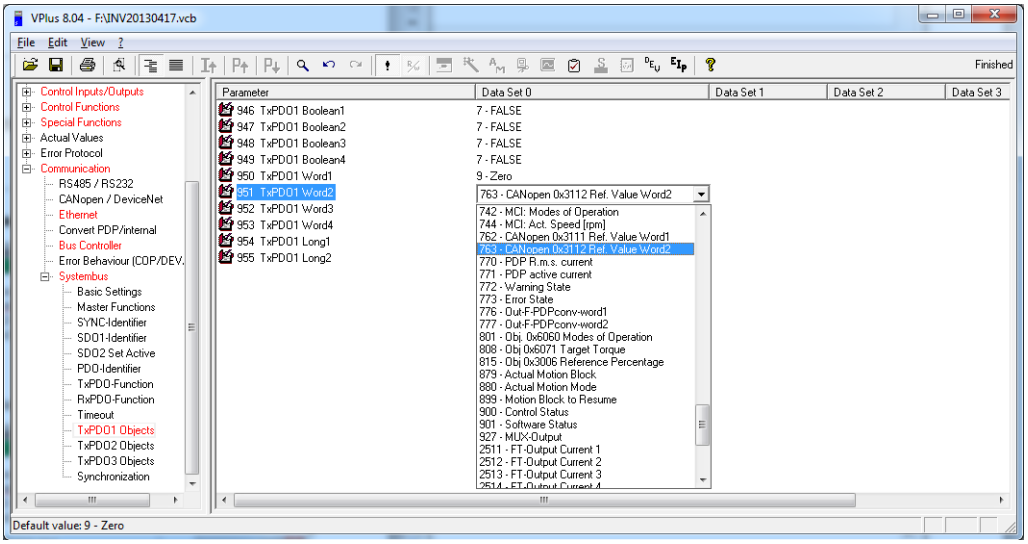
	COB ID	CB	Index	SI	Data
Read Request	601	40	11 31	00	00 00
Reply	581	4B	11 31	00	05 00
Write Access	601	2B	11 31	00	20 00
Reply	581	60	11 31	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.15 0x3112/0 Ref. Value word 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3112	0	Ref. Value word 2	Unsigned16	rw	Rx	0

Via object 0x3112 it is possible to write to a Word source like parameter *TxPDO1 Word 1* **950** of the Systembus.
The value of object 0x3112 is available as source which can be chosen by the selection of “763 - CANopen 0x3112 Ref. Value” from a parameters choice list.



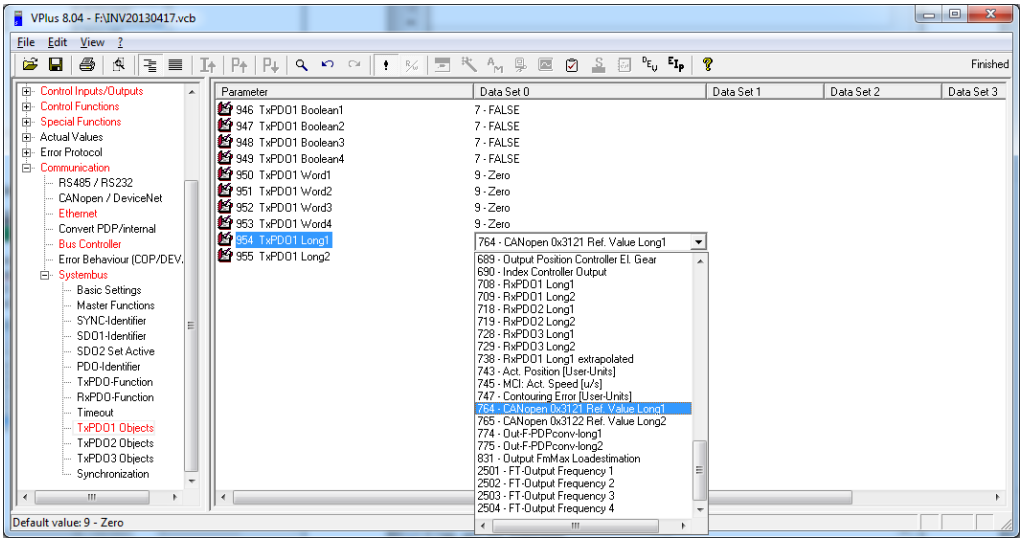
Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	12 31	00	00 00
Reply	581	4B	12 31	00	05 00
Write Access	601	2B	12 31	00	20 00
Reply	581	60	12 31	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.16 0x3121/0 Ref. Value long 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3121	0	Ref. Value long 1	Unsigned32	rw	Rx	0

Via object 0x3121 it is possible to write to a Word source like parameter *TxPDO1 Long 1* **954** of the Systembus.
The value of object 0x3121 is available as source which can be chosen by the selection of “764 - CANopen 0x3121 Ref. Value” from a parameters choice list.



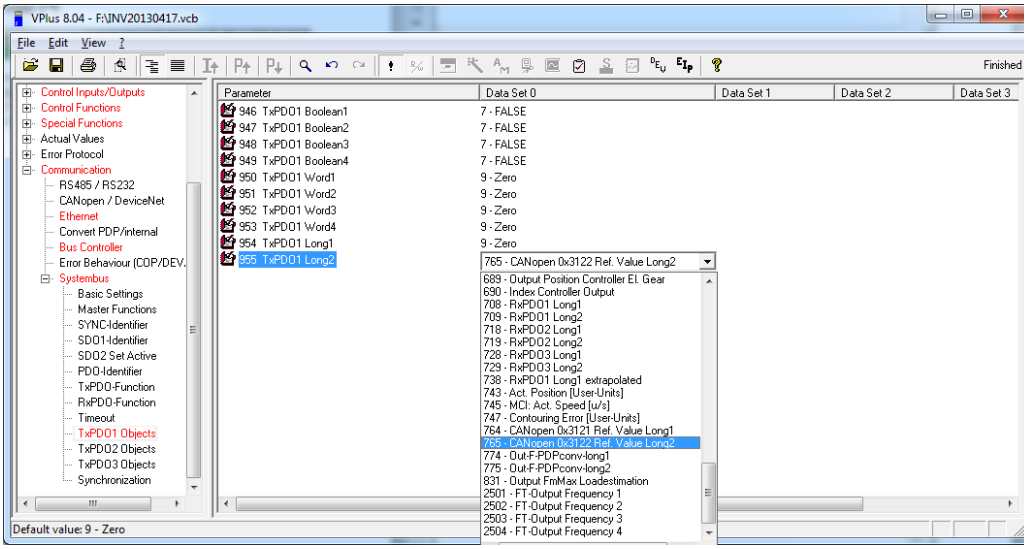
Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	21 31	00	00 00 00 00
Reply	581	43	21 31	00	05 00 00 00
Write Access	601	23	21 31	00	20 00 00 00
Reply	581	60	21 31	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.17 0x3122/0 Ref. Value long 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3122	0	Ref. Value long 2	Unsigned32	rw	Rx	0

Via object 0x3122 it is possible to write to a Word source like parameter *TxPDO1 Long 1* **954** of the Systembus.
The value of object 0x3122 is available as source which can be chosen by the selection of “765 - CANopen 0x3122 Ref. Value” from a parameters choice list.



Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	22 31	00	00 00 00 00
Reply	581	43	22 31	00	05 00 00 00
Write Access	601	23	22 31	00	20 00 00 00
Reply	581	60	22 31	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.18 0x5F10 / 0 Gear factor

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F10	0	Highest sub-index supported	Unsigned8	ro	No	3
	1	Numerator	Integer16	rw	Rx	1
	2	Denominator	Unsigned16	rw	Rx	1
	3	Resync on change	Integer16	rw	No	1

Object works in:	Object doesn't work in:
• Motion Control: o **Electronic Gear: Slave** o **Table Travel Record mode** (Electronic Gear operation)	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** • **Non motion Control** (conf. ≠ x40)

Object 0x5F10 *active motion block* is available in *Electronic Gear: Slave* mode in motion control configurations (P.30 = x**40**). The *Electronic Gear: Slave* mode is activated by object 0x6060 *modes of operation* set to **-3**.

With the Gear factors (numerator and denominator) a multiplier for the master speed can be set up. The Slave speed results in:

$$v_{Slave} = v_{Master} \times \frac{Numerator\ 0x5F10/1}{Denominator\ 0x5F10/2}$$

Limitation of acceleration when the gear factor is changed is effected via Object 0x5F10/3 *Gear Factor: Resync on change* . The slave is resynchronized with the master when the gear factor has changed. This function avoids sudden speed changes.

0x5F10 / 3 *Gear Factor: Resync on change.*	**Function**
0 - Off	Resynchronization is switched off.
1 - On	The slave is resynchronized with the master frequency when the gear factor has changed. The drive adjusts to the new frequency. The acceleration ramps set in Object 0x6083 *Profile Acceleration* is considered.

Alternatively the parameters 1123, 1124 and 1142 can be used instead of the Objects. Usage of the Objects will write the parameters in RAM (data set 5).

Object	Parameter
0x5F10/1 Gear factor Numerator	**1123** *Gear Factor Numerator*
0x5F10/2 Gear factor Denominator	**1124** *Gear Factor Denominator*
0x5F10/3 Gear factor Resync on change	**1142** *Resync. on Change of Gear-Factor*

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	10 5F	01	00 00
Reply	581	4B	10 5F	01	01 00
Write Access	601	2B	10 5F	01	02 00
Reply	581	60	10 5F	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.19 0x5F11/n...0x5F14/n Phasing 1...4

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F11	0	Highest sub-index supported	Unsigned8	ro	No	3
	1	Offset	Integer32	rw	No	0x1 0000
	2	Speed	Unsigned32	rw	No	0x5 0000
	3	Acceleration	Unsigned32	rw	No	0x5 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Electronic Gear: Slave** o **Table Travel Record mode** (Electronic Gear operation)	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel Record mode** o **Move away from Limit Switch** • **Non motion Control** (conf. ≠ x40)

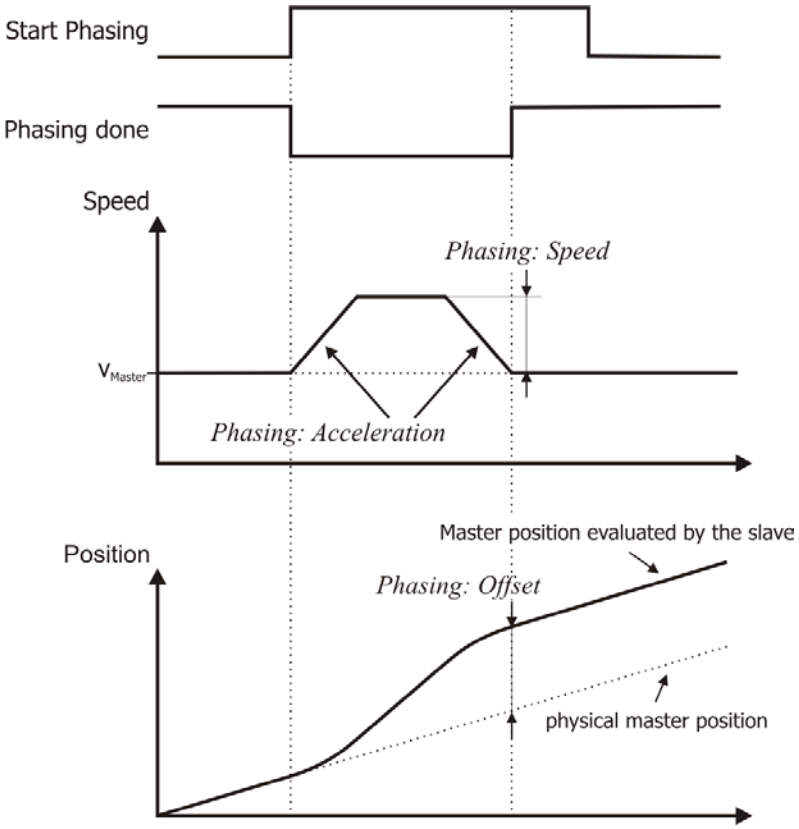
Objects 0x5F11 *Phasing 1*, 0x5F12 *Phasing 2,* 0x5F13 *Phasing 3* and 0x5F14 *Phasing 4 active motion block* is available in *Electronic Gear: Slave* mode in motion control configurations (P.30 = x**40**). The *table travel record* mode is activated by object 0x6060 *modes of operation* set to **-3**.

For better readability in the following section Object 0x5F11 is used. For Objects 0x5F12, 0x5F13 und 0x5F14 the descriptions apply analogously.

With the phasing function, the slave position is offset from the received position of the master by the value entered in 0x5F11/1 *Phasing 1: Offset*.
The function can is started via Bit 9 of the Control Word. After start, 0x5F11/2 *Phasing 1: Speed* and 0x5F11/3 *Phasing 1: Acceleration* are used until the slave position is offset from the master position by *Phasing 1: Offset*.
During Phasing the Status word bit 8 "Phasing Done" is set to "Low". As soon as the Phasing is finished or cancelled, the Bit is set to "High". After first switch-on (or after a device reset) the "Phasing Done" bit is also "Low".

The values of Objects 0x5F11/n...0x5F14/n are limited as follows:

Object		Setting	
No.	**Object**	**Min.**	**Max.**
0x5F11/1 0x5F12/1 0x5F13/1 0x5F14/1	Phasing: Offset	-2147483647 (= 0x8000 0001)	2147483647 (= 0x7FFF FFFF)
0x5F11/2 0x5F12/2 0x5F13/2 0x5F14/2	Phasing: Speed	1	2147483647 (= 0x7FFF FFFF)
0x5F11/3 0x5F12/3 0x5F13/3 0x5F14/3	Phasing: Acceleration	1	2147483647 (= 0x7FFF FFFF)



Via Objects 0x5F11, 0x5f12, 0x5F13 and 0x5F14 four different Phasing profile can be created. The Phasing Profile is selected via Control word bits 12 and 13.

Phasing select		**Phasing Profile**
Bit 13	Bit 12	
0	0	1 (0x5F11)
1	1	2 (0x5F12)
0	0	3 (0x5F13)
1	1	4 (0x5F14)

Alternatively the parameters 1125, 1126 and 1127 can be used instead of the Objects. The 4 data sets of the parameters correspond to the 4 Objects. Usage of the Objects will write the parameters in RAM (data set 6...9).

Object	Parameter	
0x5F11/1 Phasing 1: Offset	**1125.1**	*Phasing: Offset*
0x5F12/1 Phasing 2: Offset	**1125.2**	
0x5F13/1 Phasing 3: Offset	**1125.3**	
0x5F14/1 Phasing 4: Offset	**1125.4**	
0x5F11/2 Phasing 1: Speed	**1126.1**	*Phasing: Speed*
0x5F12/2 Phasing 2: Speed	**1126.2**	
0x5F13/2 Phasing 3: Speed	**1126.3**	
0x5F14/2 Phasing 4: Speed	**1126.4**	
0x5F11/3 Phasing 1: Acceleration	**1127.1**	*Phasing: Acceleration*
0x5F12/3 Phasing 2: Acceleration	**1127.2**	
0x5F13/3 Phasing 3: Acceleration	**1127.3**	
0x5F14/3 Phasing 4: Acceleration	**1127.4**	

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	11 5F	01	00 00 00 00
Reply	581	43	11 5F	01	01 00 00 00
Write Access	601	23	11 5F	01	FF 00 00 00
Reply	581	60	11 5F	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.20 0x5F15/0 In Gear Threshold

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F15	0	In Gear Threshold	Unsigned32	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Table Travel record mode** o **Electronic Gear: Slave**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** • **Non motion Control** (conf. ≠ x40)

The Status Word Bit 10 "In Gear" is set if the relative deviation between master- and slave-position is lower than the value of 0x5F15/0 *In Gear Threshold* for at least 0x5F16/0 *In Gear Time* .

If parameter 0x5F15/0 *In Gear Threshold* is set to the value zero the signal "In Gear" is set when the drive attains the master speed.

The signals "In Gear" are reset in the following occurrences:
- The relative deviation between master- and slave-position exceeds the value of 0x5F15/0 *In Gear Threshold*.
- The speed of the master drive exceeds the value of *Maximum Speed* *.

*) *Maximum speed* refers to either 0x6046/2 *Velocity max amount* or *Maximum frequency* **419**. It is set either via 0x6046/2 *Velocity max amount* [rpm] or *Maximum frequency* **419** [Hz]. *Maximum frequency* **419** is usually set up during motor commissioning.

The value range of des Object 0x5F15/0 is limited as follows:

Object		Setting	
Nr.	**Object**	**Min.**	**Max.**
0x5F15/0	In Gear Threshold	0	2147483647 (= 0x7FFF FFFF)

Alternatively parameter *"In-Gear"-Threshold* **1168** can be used instead of the Object 0x5F15/0 *In Gear Threshold*.

Object	Parameter
0x5F15/0 In Gear Threshold	**1168** *"In-Gear"-Threshold*

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	15 5F	00	00 00 00 00
Reply	581	43	15 5F	00	00 00 00 00
Write Access	601	23	15 5F	00	7F 00 00 00
Reply	581	60	15 5F	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.21 0x5F16/0 In Gear Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F16	0	In Gear Time	Unsigned16	rw	No	10

Object works in:	Object doesn't work in:
• Motion Control: o **Table Travel record mode** o **Electronic Gear: Slave**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** • **Non motion Control** (conf. ≠ x40)

The Status Word Bit 10 "In Gear" is set if the relative deviation between master- and slave-position is lower than the value of 0x5F15/0 *In Gear Threshold* for at least 0x5F16/0 *In Gear Time* .

If parameter 0x5F15/0 *In Gear Threshold* is set to the value zero the signal "In Gear" is set when the drive attains the master speed.

The signals "In Gear" are reset in the following occurrences:
- The relative deviation between master- and slave-position exceeds the value of 0x5F15/0 *In Gear Threshold*.
- The speed of the master drive exceeds the value of *Maximum Speed* *.

*) *Maximum speed* refers to either 0x6046/2 *Velocity max amount* or *Maximum frequency* **419**. It is set either via 0x6046/2 *Velocity max amount* [rpm] or *Maximum frequency* **419** [Hz]. *Maximum frequency* **419** is usually set up during motor commissioning.

The Position Controller (0x5F17 *Position Controller*) can cause a higher overall speed than *Maximum speed*. However the Position Controller doesn't affect the Signal "In Gear".

The value range of des Object 0x5F16/0 is limited as follows:

Object		Setting	
Nr.	**Object**	**Min.**	**Max.**
0x5F16/0	In Gear Time [ms]	1	65535 (= 0xFFFF)

Alternatively parameter *„In-Gear"-Time* **1169** can be used instead of the Object 0x5F16/0 *In Gear Time*.

Object	Parameter
0x5F16/0 In Gear Time	**1169** *In Gear-Time*

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	16 5F	00	00 00
Reply	581	4B	16 5F	00	0A 00
Write Access	601	2B	16 5F	00	1F 00
Reply	581	60	16 5F	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.22 0x5F17/n Position Controller

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F17	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Time Constant	Integer32	rw	No	10,00 ms
	2	Limitation	Unsigned32	rw	No	327680

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

The position controller evaluates the positioning operation (target/actual position) and tries to control the drive such that it comes as close as possible to the specifications. For this purpose, an additional frequency is calculated for compensation of position deviations. By setting the corresponding parameter, this frequency can be limited. The parameter settings of the position controller determine how quick and to what extent position deviations are to be compensated.
Via the *Position Controller:Time Constant* , you can define the maximum time in which the position deviation is to be compensated.
Via parameter *Position Controller:Limitation*, you can define to which value the speed is limited for compensation of the position deviation.

NOTE

The Output of the Position Controller is not limited by 0x6046/2 *Velocity max amount* (or *Maximum frequency* **419**). The *Maximum speed** limits the value of the Motion Profile generation. Caused by the addition of the Profile generator reference speed and the output of the Position Controller higher frequencies than *Maximum speed** can occur.

*Maximum speed** and *Limitation* **1118** must be set for fitting values during the commissioning.
Chapter 16.5 contains conversion formulas between Hz, rpm and u/s.
Bonfiglioli Vectron recommends:

- Set *Maximum speed** to 90 % of the mechanical rated speed and

the *Limitation* **1118** of the Position Controller to the value corresponding to 10 % of the Maximum frequency.
*) *Maximum speed* refers to either 0x6046/2 *Velocity max amount* or *Maximum frequency* **419**. It is set either via 0x6046/2 *Velocity max amount* [rpm] or *Maximum frequency* **419** [Hz]. *Maximum frequency* **419** is usually set up during motor commissioning.

The values of Objects 0x5F17/n are limited as follows:

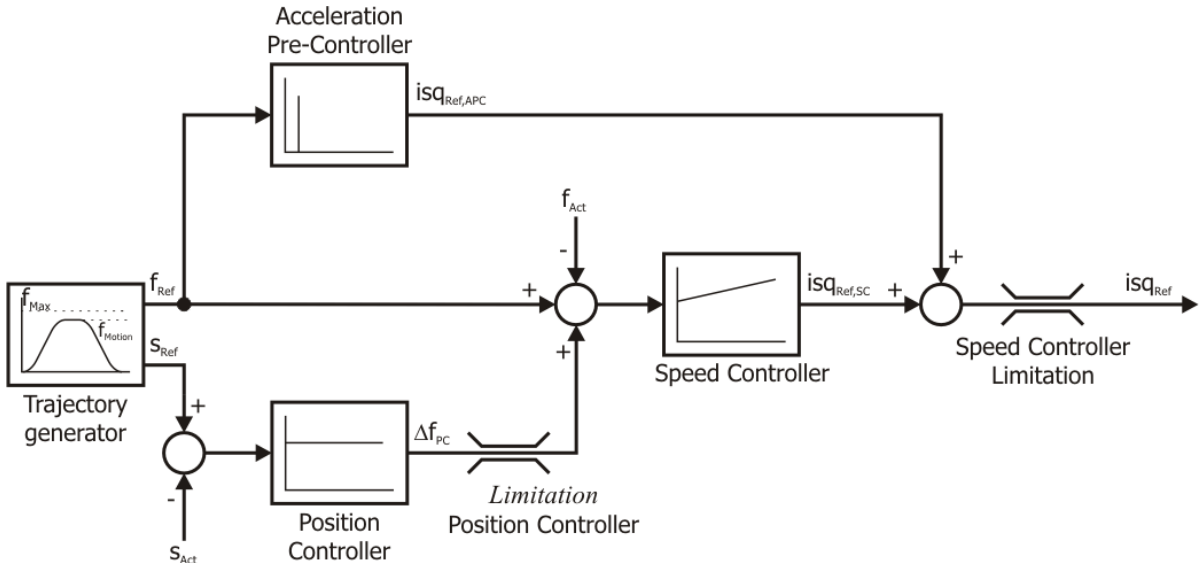
Object		Setting	
Nr.	**Object**	**Min.**	**Max.**
0x5F17/1	Position Controller: Time Constant	1.00 ms	300.00 ms
0x5F17/2	Position Controller: Limitation	0	2147483647 (= 0x7FFF FFFF)

Alternatively the parameters **1104** and **1118** can be used instead of the Objects.

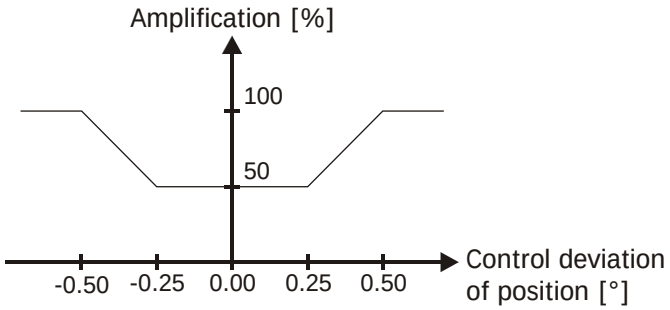
Object	Parameter
0x5F17/1 Position Controller: Time Constant	**1104** *Time Constant*
0x5F17/2 Position Controller: Limitation	**1118** *Limitation*

Example:
Position deviates by 1 motor shaft revolution, time constant is set to 1 ms. The position controller will increase the motor frequency by 1000 Hz in order to compensate the position deviation. Parameter *Limitation* **1118** must be set accordingly.

Controller block diagram



In order to avoid oscillations of the drive while it is at standstill, amplification is reduced to 50 % of the parameterized value for small position deviations



The following behaviour may indicate that the controller parameters are not configured properly:

- drive is very loud
- drive vibrates
- frequent contouring errors
- inexact control

For the setting options of other control parameters, e.g. speed controller and acceleration pilot control, refer to the operating instructions of the frequency inverter.

Optimize the settings in actual operating conditions, as control parameters for speed controller and acceleration pilot control depend on actual load. Optimize with different load types to obtain a good control behaviour in all situations.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	17 5F	01	00 00 00 00
Reply	581	4B	17 5F	01	E8 03 00 00
Write Access	601	2B	17 5F	01	D0 07 00 00
Reply	581	60	17 5F	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.23 0x5F18/0 M/S Synchronization Offset

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5F18	0	M/S Synchronization Offset	Integer32	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Table Travel record mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** • **Non motion Control** (conf. ≠ x40)

The M/S Synchronization Offset can be used in the electronic gear functionality to align the Slave drive to the absolute position of the master drive.
Comply with chapter 14.4.10.1 "Master/Slave Position Correction".

NOTE

When using this functionality master drive and slave drive have to use the same mechanical characteristics (i.e. gear transmission ratios) and use the same reference system.

The values of Object 0x5F18/0 are limited as follows:

Object		Setting	
No.	**Object**	**Min.**	**Max.**
0x5F18/0	M/S Synchronization Offset	-2147483647 (= 0x8000 0001)	2147483647 (= 0x7FFF FFFF)

Alternatively parameter 1284 can be used instead of the Objects.

Object	Parameter
0x5F18/0 M/S Synchronization Offset	**1284** *M/S Synchronization Offset*

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	18 5F	00	00 00 00 00 00 00 00 00
Reply	581	42	18 5F	00	E8 03 00 00 00 00 00 00
Write Access	601	22	18 5F	00	D0 07 00 00 00 00 00 00
Reply	581	60	18 5F	00	00 00 00 00 00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.24 0x5FF0 / 0 Active motion block

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5FF0	0	Active motion block	Unsigned8	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **Table Travel record mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x5FF0 *active motion block* is only available in *table travel record* mode in motion control configurations (P.30 = x**40**). The *table travel record* mode is activated by object 0x6060 *modes of operation* set to **-1**.

The data of *active motion block* displays the active motion block number in *table travel record* mode. The Object refers to parameter *Actual motion block* **1246**. Refer to the application manual "Positioning" for the usage of the motion blocks.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	F0 5F	00	00
Reply	581	4B	F0 5F	00	01

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.4.25 0x5FF1/0 Motion block to resume

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x5FF1	0	Motion block to resume	Unsigned8	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **Table Travel record mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x5FF1 *motion block to resume* is only available in *table travel record* mode in motion control configurations (P.30 = x**40**). The *table travel record* mode is activated by object 0x6060 *modes of operation* set to **-1**.

The data of *motion block to resume* displays the motion block to resume number in *table travel record* mode. The Object refers to parameter *Motion block to resume* **1249**. Refer to the application manual "Positioning" for the usage of the motion blocks.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	F1 5F	00	00
Reply	581	4B	F1 5F	00	01

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5 Device Profile Objects (0x6nnn)

12.5.1 0x6007/0 Abort Connection option code

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6007	0	Abort Connection option code	Integer16	rw	No	1

Object *abort connection option code* specifies the operational behaviour of the frequency inverter if the bus connection fails due to BusOff, guarding, heartbeat, SYNC error, RxPDO length error or NMT state change (leaving NMT state operational).

Depending on the setting of parameter *Local/Remote* **412** the reaction of the setting of Object 0x6007 changes like displayed in the next table.

Object 0x6007/0		
Operation mode	**Function with "Control via Statemachine"**	**Function with "Control via Contacts"**
0 - No reaction	Operating point is maintained.	Operating point is maintained.
1 - Error (**factory setting**)	Device state machine changes to state "fault" immediately.	Device state machine changes to state "fault" immediately.
2 - Switch-off	Device state machine processes command '*disable voltage*' and changes to state "switch on disabled"	
3 - Quick stop	Device state machine processes command '*quick stop*' and changes to state "switch on disabled"	
-1 (minus 1) - Ramp stop + Error	Device state machine processes command '*disable operation*' and changes to state "fault" after the drive is stopped	
-2 (minus 2) - Quick stop + Error	Device state machine processes command '*quick stop*' and changes to state "fault" after the drive is stopped	

NOTE

The object *abort connection option code* corresponds to the inverter parameter *Bus Error behaviour* **388**.

The settings of *Bus Error behaviour* **388** = -2...3 are evaluated depending on parameter *Local/Remote* **412**.

No.	Object	Min.	Max.
0x6007/0	Abort Connection option code	-2 (=0xFFFE)	3

Bus Error behaviour **388**	**0x6007**
0	0
1	1
2	2
3	3
4	-1
5	-2

Writing *Bus Error behaviour* **388** or writing object 0x6007 has the same effect.

If object 0x6007 was written and then a save parameters command (object 0x1010) processed, the value of 0x6007 is stored in non volatile memory. After the next power on of the inverter the previously set value for 0x6007 is reactivated and overwrites the setting of *Bus Error behaviour* **388**.

Occurring errors are described in detail in chapter 16.4 “Fault messages”.

NOTE

The behaviour for fault reset corresponds to object 0x1029 *Error Behaviour*. Depending on the setting of object *Error behaviour,* the NMT state may change (leaving NMT state operational). In this case the NMT state must be set to operational again before a fault reset command is accepted by the inverter.

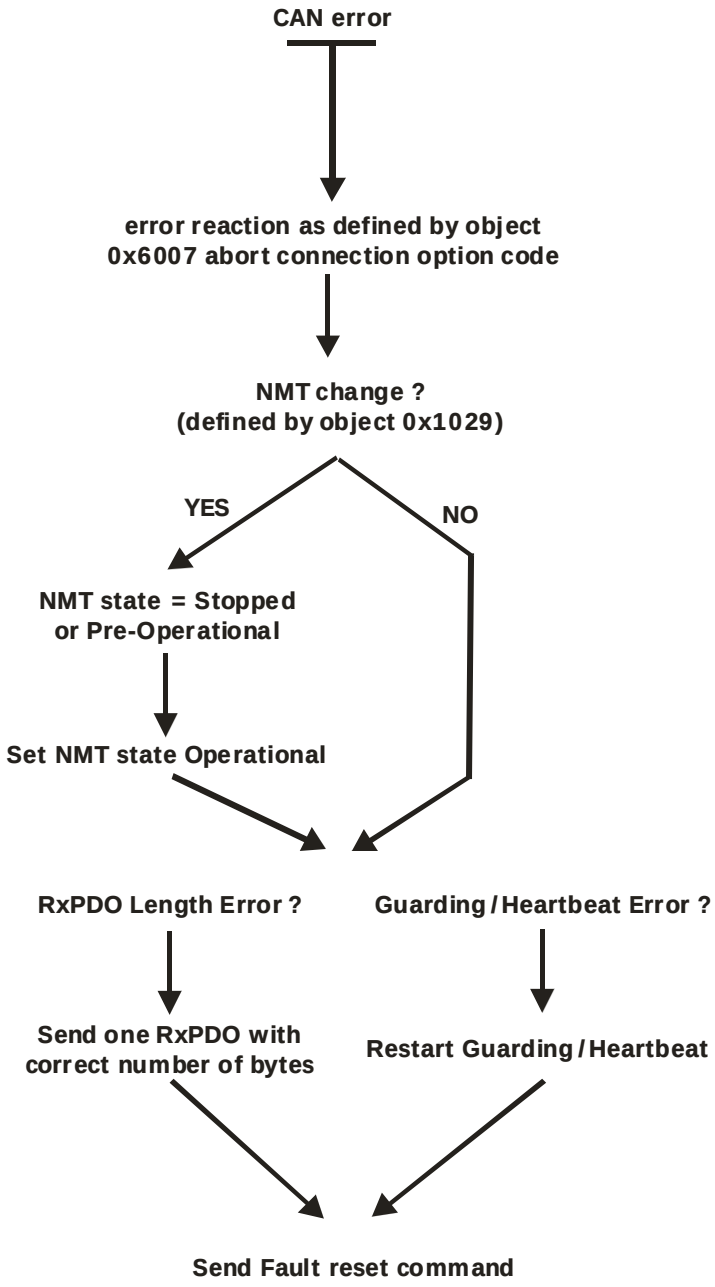
In addition, the error reason must also be reset. E.g. for a guarding error, guarding must be started again before a fault reset command is accepted by the inverter. In the case of an RxPDO length error, an RxPDO frame with the correct number of bytes must be received before new RxPDO data is accepted.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	07 60	00	00 00
Reply	581	4B	07 60	00	01 00
Write Access	601	2B	07 60	00	FE FF
Reply	581	60	07 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

Typical recovery sequence after CAN error:



⚠ WARNING

With the setting of object 0x6007 *abort connection option code* = 0 (no reaction), the inverter does not react to any CAN error and remains in its current state (e. g. drive running).

It is strongly recommended to use a setting for object 0x6007 *abort connection option code* that forces the inverter into the Fault state (setting = 1, -1 or -2).

12.5.2 0x603F/0 Error code

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x603F	0	Error code	Unsigned16	ro	No	

The object *error code* is used to store the last error that occurred.

In DS402, a large number of possible error codes are specified. The following list shows the relationship between the error code displayed internally by the frequency inverter and on the KP500 control unit, and the error secured in object *error code*.

Error reports				
Inverter Error		**DS402 Error code**		**Meaning**
F00	xx	00	00	No error has occurred
Overload				
F01	xx	23	10	Frequency inverter has been overloaded
Case				
F02	xx	42	10	Case temperature outside the temperature limits
Inside				
F03	xx	41	10	Inside temperature outside the temperature limits
Motor connection				
F04	xx	43	10	Motor temperature too high or sensor defective
Output current				
F05	xx	23	40	Motor phase current above the current limit
DC link voltage				
F07	xx	32	10	DC link voltage outside the voltage range
Electronic voltage				
F08	xx	51	11	Electronic voltage outside the voltage range
Motor connection				
F13	xx	23	30	Earth fault on frequency inverter output
Generic error				
Fyy	xx	10	00	Other error reports

If the value 1000 (= generic error) appears as the DS402 error code, then the inverter error code can be read via parameter *Current error* **260** (unsigned16). This parameter contains the error code in the product-internal format.

The assignment table of the error code can be taken from the operating instructions.

In the emergency message, the error code of the frequency inverter is transmitted in bytes 4 ... 7 and the DS402 error code in bytes 0 and 1. Please refer to object 0x1014 *COB-ID Emergency Message* for further explanations.

Example:

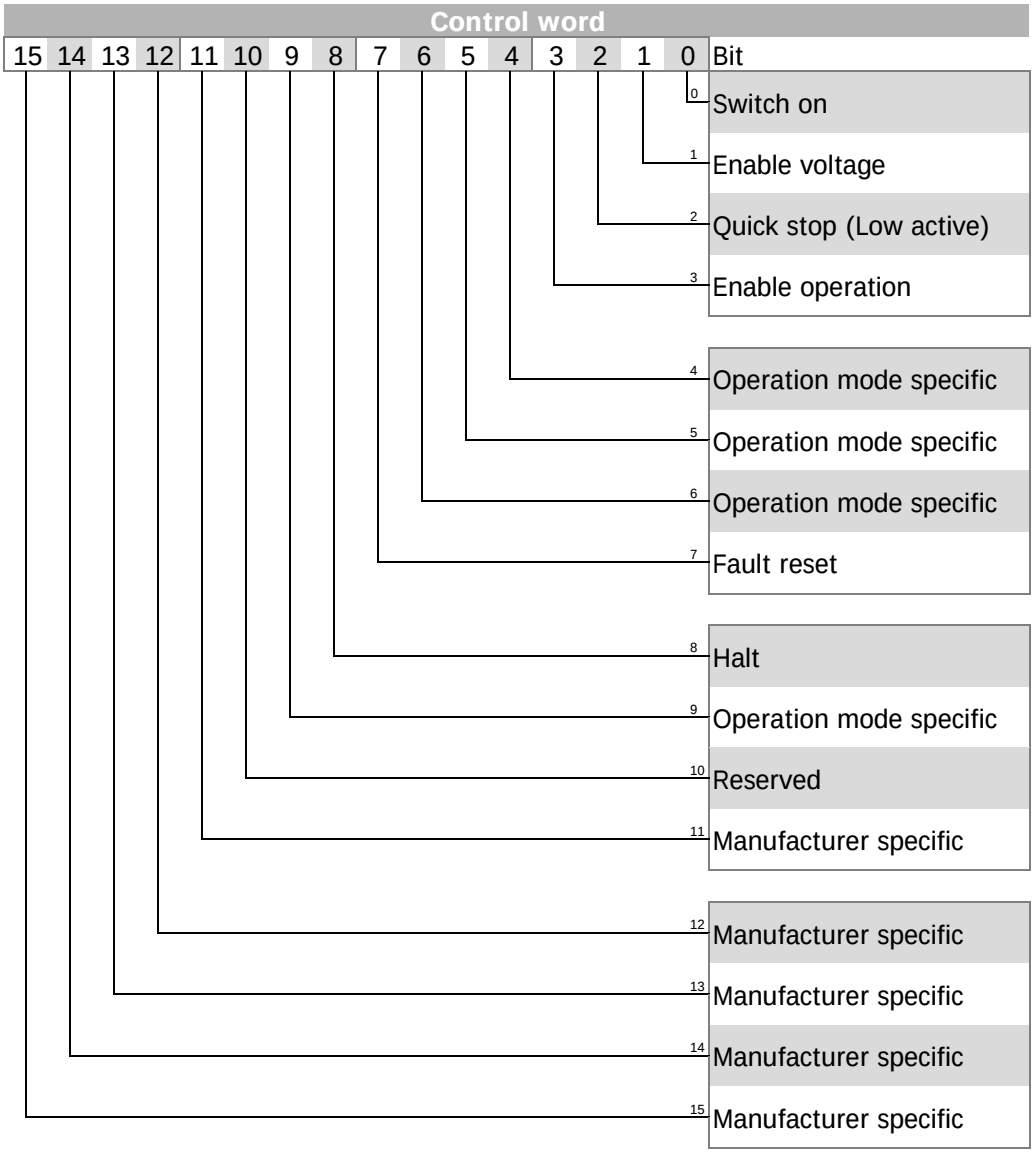
	COB ID	CB	Index	SI	Data
Read Request	601	40	3F 60	00	00 00
Reply	581	4B	3F 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.3 0x6040/0 Control word

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6040	0	Control word	Unsigned16	rw	Rx	0

Object 0x6040/0 *Control word* is relevant to the inverter remote state machine whenever parameter *LocalRemote* **412** is set to **1** (remote state machine).



Bits 4, 5, 6 and 9 ... 15 are used in motion control configurations (p.30 = x**40**) only.
See chapter 14 “Inverter Control” and 16.1.1 “Control Word overview”.

Example:

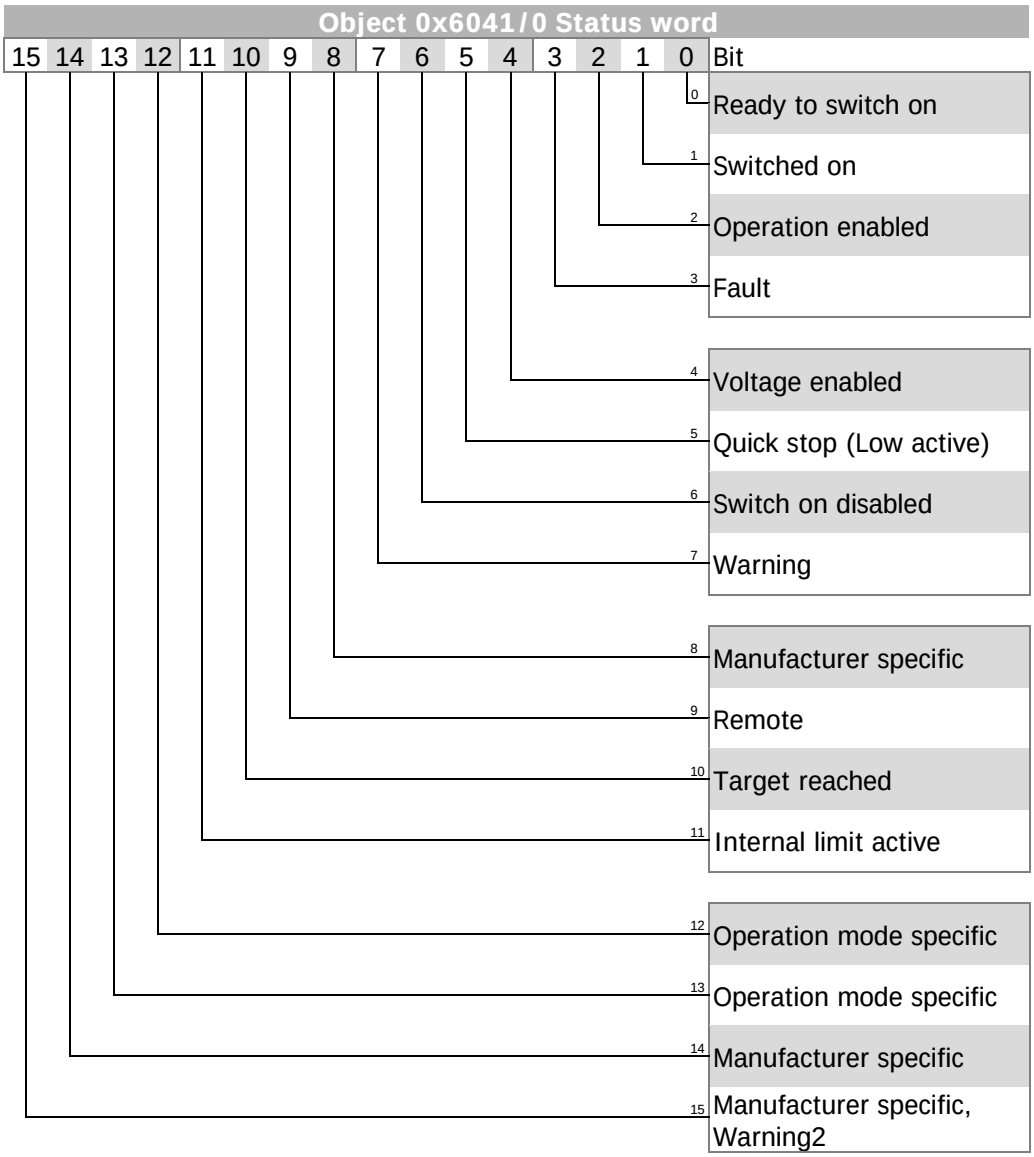
	COB ID	CB	Index	SI	Data
Read Request	601	40	40 60	00	00 00
Reply	581	4B	40 60	00	01 00
Write Access	601	2B	40 60	00	06 00
Reply	581	60	40 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.4 0x6041/0 Status word

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6041	0	Status word	Unsigned16	ro	Tx	

Object 0x6041/0 *Status word* displays the actual state of the inverter.



Bits 8, 12, 13 and 14 *operation mode specific* are used in motion control configurations (p.30 = x**40**) only.
See chapter 14 “Inverter Control” and chapter 0 “ Status Word overview”.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	41 60	00	00 00
Reply	581	4B	41 60	00	31 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.5 0x6042/0 Target velocity [rpm]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6042	0	Target velocity	Integer16	rw	Rx	0

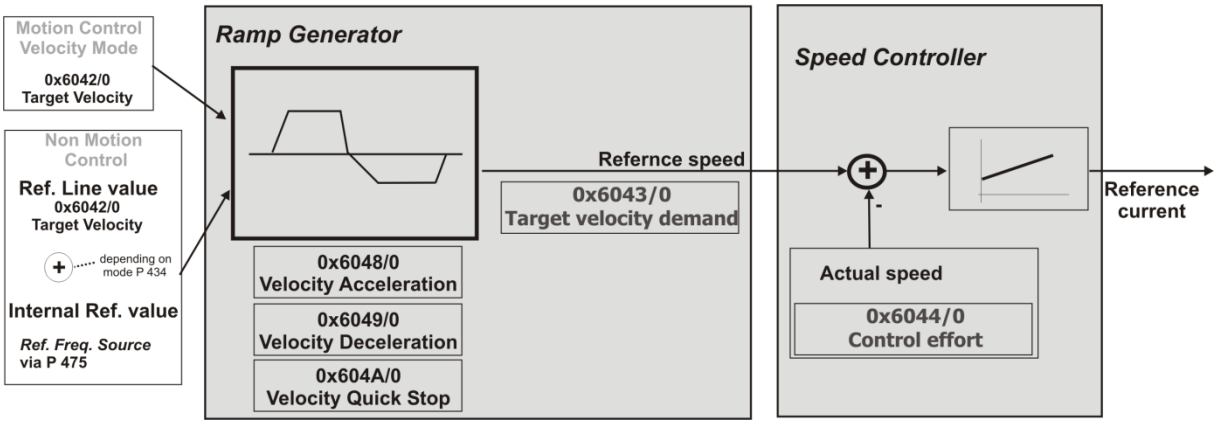
Object works in:	Object doesn't work in:
• Motion Control: o **Velocity mode** • **Non motion Control** (conf. ≠ x40)	• Motion Control: o **Table Travel record mode** o **Profile Positioning mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**

Object *target velocity* is the speed reference value for the frequency inverter. *Target velocity* is interpreted as a speed with the unit RPM. The inverter's internal reference frequency is calculated from the target velocity in RPM taking into account parameter *No. of Pole Pairs* **373**.

The parameter *No. of Pole Pairs* **373** has four different data sets. In motion control applications (configuration = x40) only the data set 1 is used.
Non motion control applications (configuration ≠ x40) sometimes have more than one motor connected to the inverter (only one at a time, switched over by contactor). These motors may have a different no. of pole pairs. The entry in *No. of Pole Pairs* **373** is then different in the four data sets. After change-over of the motor, the object *target velocity* must be written at least once in order to recalculate the internal reference frequency of the inverter using the correct no. of pole pairs.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6042	Target velocity	-32768	32767

In Non motion Control configurations (conf. ≠ x40) the target velocity reference value is product-internally connected to the **Reference line value**. This reference value is combined with the internal reference frequency value from the frequency reference value channel in the input of the ramp function (see chapter 14.3.3 "Reference value / actual value").



Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	42 60	00	00 00
Reply	581	4B	42 60	00	00 00
Write Access	601	2B	42 60	00	DC 05
Reply	581	60	42 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.6 0x6043/0 Target velocity demand

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6043	0	Target velocity demand	Integer16	ro	Tx	

Object *target velocity demand* is the output value of the ramp function in RPM. The object has the same notation as the object *target velocity* and can be read as an actual value. For calculating *target velocity demand* the parameter *No. of Pole Pairs* **373** is taken into account in the same way as described for object target velocity.

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	43 60	00	00 00
Reply	581	4B	43 60	00	AB 01

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.7 0x6044/0 Control effort

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6044	0	Control effort	Integer16	ro	Tx	

Object *control effort* is the actual speed of the drive in RPM. The object has the same notation as the object *target velocity* and can be read as an actual value. For calculating *control effort* the parameter *No. of Pole Pairs* **373** is taken into account in the same way as described for object target velocity.

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	44 60	00	00 00
Reply	581	4B	44 60	00	DE 01

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.8 0x6046/n Velocity min max amount

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6046	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Velocity min amount (RPM)	Unsigned32	rw	No	See text
	2	Velocity max amount (RPM)	Unsigned32	rw	No	See text

Object works in:	Object doesn't work in:
• Motion Control: o **All modes** • **Non motion Control** (conf. ≠ x40)	

Object velocity min max amount comprises the sub-index 1 = velocity min amount and sub-index 2 = velocity max amount.

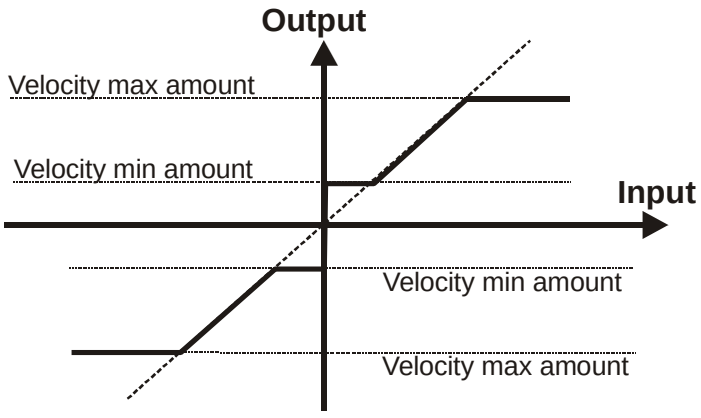
The unit of 0x6046/1 *velocity min amount* is in RPM (positive values only). Writing to object 0x6046/1 *velocity min amount* automatically generates a write command to parameter *Minimum Frequency* **418** into RAM (data set 5, all data sets in RAM only). The value of 0x6046/1 *velocity min amount* is converted internally to a frequency value, taking into account parameter *No. of Pole Pairs* **373** (in data set 1 !).

The unit of 0x6046 *velocity max amount* is in RPM (positive values only). Writing to object 0x6046/2 *velocity max amount* automatically generates a write command to parameter *Maximum Frequency* **419** into RAM (data set 5, all data sets in RAM only). The value of 0x6046/2 *velocity max amount* is converted internally to a frequency value, taking into account parameter *No. of Pole Pairs* **373** (in data set 1 !).

The default values depend on the used motor settings.

If the input reference value of object 0x6042 *target velocity* is less than the object value 0x6046/1 *velocity min amount* or greater than 0x6046/2 *velocity max amount*, then 0x6042 *target velocity* is limited accordingly.

No.	Object	Min.	Max.
0x6046/1	Velocity min amount (RPM)	1	32767 (= 0x7FFF)
0x6046/2	Velocity max amount (RPM)	1	32767 (= 0x7FFF)



If objects 0x6046/1 or 0x6046/2 were written and then a save parameters command (object 0x1010) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of parameters *Minimum Frequency* **418** and *Maximum Frequency* **419**.

In Positioning applications the overall speed can fall below or exceed the limits defined by Minimum and Maximum frequency due to the influence of the Position controller. The output of the Position Controller can be limited by *Limitation* **1118.**

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	46 60	01	00 00 00 00
Reply	581	43	46 60	01	00 00 00 00
Write Access	601	23	46 60	01	DC 05 00 00
Reply	581	60	46 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.9 0x6048/n Velocity acceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6048	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (RPM)	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

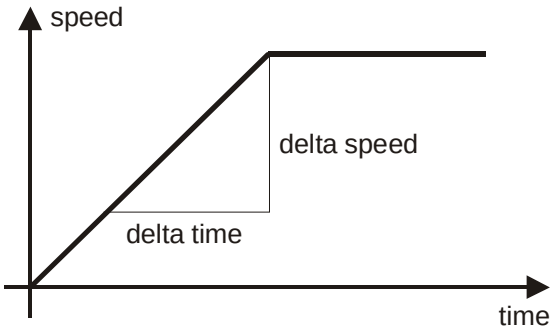
Object works in:	Object doesn't work in:
• Motion Control: o **Velocity mode** • **Non motion Control** (conf. ≠ x40)	• Motion Control: o **Table Travel record mode** o **Profile Positioning mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**

The acceleration in **velocity mode** is set with object *velocity acceleration*. The object *velocity acceleration* consists of *delta speed* in RPM and *delta time* in seconds.

The gradient of the frequency in the acceleration period is written to parameters *Acceleration (clockwise)* **420** and *Acceleration (anti-clockwise)* **422** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.420 and p.422 are converted internally to a frequency/sec value, taking into account parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects delta time or delta speed.

No.	Object	Min.	Max.
0x6048/1	Delta speed (RPM)	1	32767 (= 0x7FFF)
0x6048/2	Delta time (sec)	1	65535 (= 0xFFFF)



If objects 0x6048/1 or 0x6048/2 were written and then a save parameters command (object 0x1010) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of parameters *Acceleration Clockwise* **420** und *Acceleration Counterclockwise* **422**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	48 60	01	00 00 00 00
Reply	581	43	48 60	01	96 00 00 00
Write Access	601	23	48 60	01	50 50 00 00
Reply	581	60	48 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.10 0x6049/n Velocity deceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6049	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (RPM)	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

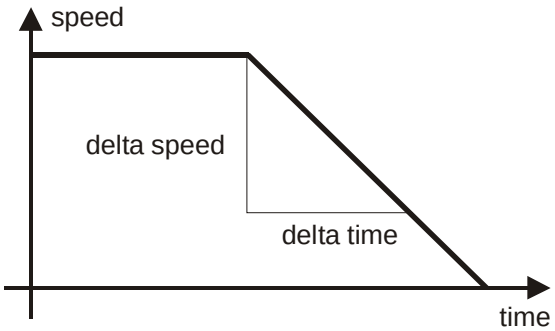
Object works in:	Object doesn't work in:
• Motion Control: o **Velocity mode** • **Non motion Control** (conf. ≠ x40)	• Motion Control: o **Table Travel record mode** o **Profile Positioning mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**

The deceleration in **velocity mode** is set with object *velocity deceleration*. The object *velocity deceleration* consists of *delta speed* in rpm and *delta time* in seconds.

The gradient of the frequency in the deceleration period is written to parameters *Deceleration (clockwise)* **421** and *Deceleration (anti-clockwise)* **423** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.421 and p.423 are converted internally to a frequency/sec value, taking into account the parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects delta time or delta speed.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6049/1	Delta speed (RPM)	1	32767 (= 0x7FFF)
0x6049/2	Delta time (sec)	1	65535 (= 0xFFFF)



If objects 0x6049/1 or 0x6049/2 were written and then a save parameters command (object 0x1010) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of parameters *Deceleration Clockwise* **421** und *Deceleration Counterclockwise* **423**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	49 60	01	00 00 00 00
Reply	581	43	49 60	01	96 00 00 00
Write Access	601	23	49 60	01	40 50 00 00
Reply	581	60	49 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.11 0x604A/n Velocity quick stop

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x604A	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (RPM)	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

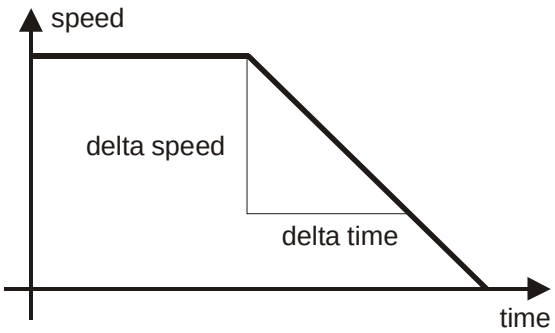
Object works in:	Object doesn't work in:
• Motion Control: o **Velocity mode** • **Non motion Control** (conf. ≠ x40)	• Motion Control: o **Table Travel record mode** o **Profile Positioning mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**

The quick stop deceleration in velocity mode is set with object *velocity quick stop*. Object *velocity quick stop* consists of *delta speed* in RPM and *delta time* in seconds.

The gradient of the frequency in the deceleration period is written to parameters *Emergency Stop (clockwise)* **424** and *Emergency Stop (anti-clockwise)* **425** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.424 and p.425 are converted internally to a frequency/sec value, taking into account the parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects delta time or delta speed.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x604A/1	Delta speed (min^{-1})	1	32767 (= 0x7FFF)
0x604A/2	Delta time (sec)	1	65535 (= 0xFFFF)



If objects 0x604A/1 or 0x604A/2 were written and then a save parameters command (object 0x1010) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of *Emergency Stop Clockwise* **424** und *Emergency Stop Counterclockwise* **425**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	4A 60	01	00 00 00 00
Reply	581	43	4A 60	01	96 00 00 00
Write Access	601	23	4A 60	01	20 50 00 00
Reply	581	60	4A 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.12 0x6060/0 Modes of operation

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6060	0	Modes of operation	Integer8	wo	Rx	2

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

With object *modes of operation*, the designated operation mode of the inverter is set. Depending on the used configuration of the inverter, there are different choices feasible.

Available values for *modes of operation* with inverter in motion control configuration (*Configuration* **30** = x**40** and **412** *Local/Remote* = „1 – Control via Statemachine"):

Modes of operation	
Dec Hex.	**Mode**
1 0x01	– Profile position mode
2 0x02	– Velocity mode **(Default)**
3 0x03	– Profile velocity mode
6 0x06	– Homing mode
7 0x07	– Interpolated position mode
8 0x08	– Cyclic sync position mode
9 0x09	– Cyclic sync velocity mode
-1 0xFF	– Table travel record (manufacturer specific mode)
-2 0xFE	– Move away from Limit switch (manufacturer specific mode)
-3 0xFD	– Electronic Gear: Slave (manufacturer specific mode)

Object 0x6060 *modes of operation* is limited like described in the table.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6060/0	Modes of operation	-3 | 0xFD	9

Available value for *modes of operation* with inverter in non motion control configuration (*Configuration* **30** ≠ x**40** or **412** *Local/Remote* = „1 – Control via Statemachine"):

Modes of operation
2 – velocity mode

The inverter in non motion control configuration ignores all settings other than "**2**". When accessing via SDO, an SDO fault message is generated, that prompts the invalid value.
For further information see chapter 14 "Inverter Control".

Example:					
	COB ID	CB	Index	SI	Data
Write Access	601	2F	60 60	00	01
Reply	581	60	60 60	00	00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.13 0x6061/0 Modes of operation display

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6061	0	Modes of operation display	Integer8	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40): Value always "2"

Object 0x6061 *modes of operation display* acknowledges the previously set value of *modes of operation* by displaying the same value as *modes of operation*.

After setting 0x6060 *modes of operation*, the PLC must wait for this acknowledgement before sending any other command to the inverter.

For further information see chapter 14 "Inverter Control".

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	61 60	00	00
Reply	581	4F	61 60	00	02

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.14 0x6064/0 Position actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6064	0	Position actual value	Integer32	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

Object 0x6064 *position actual value* represents the actual value of the position measurement device in user units.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*. The value is the same like stated in parameter *Actual Position* **1108**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	64 60	00	00
Reply	581	4F	64 60	00	02

12.5.15 0x6065/0 Following error window

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6065	0	Following error window	Unsigned32	rw	No	0xFFFF FFFF

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

Object 0x6065 *following error window* is used to set the threshold of a device warning when the following error becomes too big.

In the application manual "Positioning", the term "Contouring error" is used instead of the CANopen® term "Following error".

Object 0x6065 *following error window* defines a range of tolerated position values symmetrical to the *position demand value* defined in user units.
The valid value range of object 0x6065/0 *following error window* is 0 ... 0x7FFF FFFF (2^{31}-1). Writing a value of 0x8000 0000 (2^{31})... 0xFFFF FFFE (2^{32}-2) results in an SDO abort (value range).

If the value of the *following error window* is set to 0xFFFF FFFF (2^{32}-1) OR 0, the *following error window* is switched off.
The actual following error is displayed in object 0x60F4 Following error actual value.

The warning is triggered if the Following error window was exceeded for the time specified in Object 0x6066 *following error time out*. No device fault is triggered.

Writing to object *following error window* automatically generates a write command to contouring error parameter *Warning Threshold* **1105** (data set 5, all data sets in RAM only !).

If object 0x6065/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of *Warning Threshold* **1105**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	65 60	00	00 00 00 00
Reply	581	43	65 60	00	FF FF FF FF
Write Access	601	23	65 60	00	03 E8 00 00
Reply	581	60	65 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.16 0x6066/0 Following error time out

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6066	0	Following error time out	Unsigned16	rw	No	0xA (=10)

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

When a following error (Object 0x6065 *following error window*) occurs longer than the defined value of object 0x6066 *following error time out* given in milliseconds, the corresponding bit in the Status word (bit 13 *following error*) is set to one. No device fault is triggered.

Writing to object *following error time out* automatically generates a write command to parameter *Contouring Error Time* **1119** (data set 5, all data sets in RAM only !).

If object 0x6066/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of *Contouring Error Time* **1119**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	66 60	00	00 00
Reply	581	4B	66 60	00	0A 00
Write Access	601	2B	66 60	00	03 E8
Reply	581	60	66 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

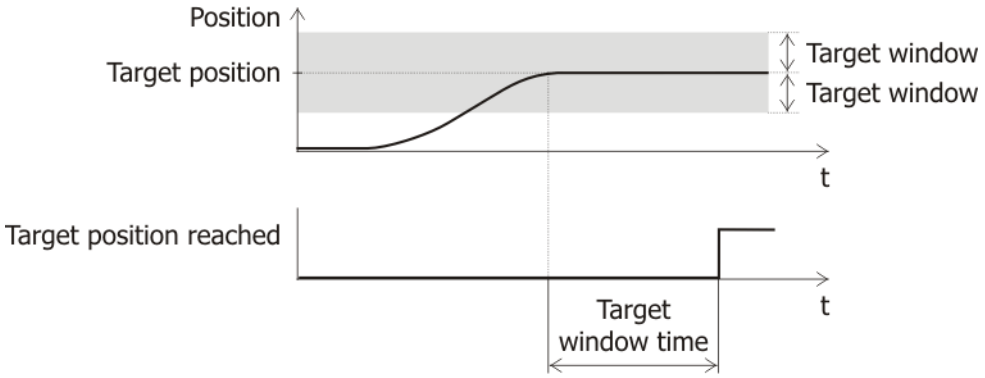
12.5.17 0x6067/0 Position window

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6067	0	Position window	Unsigned32	rw	No	0xFFFF FFFF

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

The signal "target position reached" can be changed in accuracy with Object 0x6067 *position window* for the modes which use Status Word Bit 10 "Target reached" as "Target Position reached" like "Profile Positioning Mode" and "Table Travel Record Mode".
Object 0x6067 *position window* defines a symmetrical range of accepted positions relative to the target position in user units. If the actual value of the position measurement device is within the position window, the target position is regarded as reached. "Target reached" is displayed in Bit 10 of the status word. The actual position must be inside the position window during the time specified in Object 0x6068 *position window time*.
If the actual position drifts outside the target window or if a new target position is set, the "Target reached" Bit is reset until the position and time conditions are met again.

The valid value range of object 0x6067/0 *position window* is 0 ... 0x7FFF FFFF (2^{31}-1). Writing a value of 0x8000 0000 (2^{31})... 0xFFFF FFFE (2^{32}-2) results in an SDO abort (value range).
If the value of *position window* is set to 0xFFFF FFFF (2^{32}-1) OR 0, the position window control is switched off.



Writing to object *position window* automatically generates a write command to parameter *Target Window* **1165** (data set 5, all data sets in RAM only !).

If object 0x6067/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of *Target Window* **1165**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	67 60	00	00 00 00 00
Reply	581	43	67 60	00	FF FF FF FF
Write Access	601	23	67 60	00	10 27 00 00
Reply	581	60	67 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.18 0x6068/0 Position window time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6068	0	Position window time	Unsigned16	rw	No	0xA (=10)

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

When the actual position is within the *position window* during the defined *position window time* (given in milliseconds), then the corresponding bit in the Status word (bit 10 *target reached*) is set to one. This is considered in Modes that use Status Word Bit 10 "Target reached" as "Target Position reached" like "Profile Positioning Mode" and "Table Travel Record Mode".

Writing to object *position window time* automatically generates a write command to parameter *Target Window Time* **1166** (data set 5, all data sets in RAM only !).

If object 0x6068/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of *Target Window Time* **1166**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	68 60	00	00 00
Reply	581	4B	68 60	00	0A 00
Write Access	601	2B	68 60	00	C8 00
Reply	581	60	68 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.19 0x606C/0 Velocity actual value [u/s]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606C	0	Velocity actual value	Integer32	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

The actual velocity value in [u/s] is displayed.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	6C 60	00	00 00 00 00
Reply	581	4B	6C 60	00	0A 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.20 0x606D/0 Velocity Window

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606D	0	Velocity Window	Unsigned16	rw	No	1000

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x606D *Velocity window* is used to define the threshold of Bit 10 "Target reached" of the Status word in Profile Velocity mode.

Object 0x606D *Velocity window* defines the symmetric range around the value of Object 0x60FF *Target Velocity* in user units/s.
Bit 10 "Target reached" is set in the Status word when the difference between 0x60FF *Target Velocity* and 0x606C *Velocity Actual value* is smaller than the 0x606D *Velocity Window* for a longer time than 0x606E *Velocity Window Time*.

The value range of Object 0x606D/0 *Velocity Window* is 0 ... 65535 u/s.
If the value of 0x606D/0 *Velocity Window* is set to 0, bit 10 "Target reached" of the Status word is only set with the exact equality of actual speed and reference speed. Bonfiglioli Vectron recommends setting the value large enough to get a reliable status information of Bit 10 "Target reached".

Writing to object 0x606D/0 *Velocity Window* automatically generates a write command to parameter *Velocity Window* **1276** (data set 5, all data sets in RAM only !).

If object 0x606D/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Velocity Window* **1276**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	6D 60	00	00 00
Reply	581	4B	6D 60	00	E8 03
Write Access	601	2B	6D 60	00	64 00
Reply	581	60	6D 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.21 0x606E/0 Velocity Window Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606E	0	Velocity Window time	Unsigned16	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x606E *Velocity window Time* defines the time, for which at least reference velocity and Actual velocity must be similar enough to set Bit 10 "Target reached" of the Status word. The similarity ("Hysteresis") is defined via 0x606D *Velocity Window*.

0x606D *Velocity window* defines the symmetric range around the value of Object 0x60FF *Target Velocity* in user units/s.
Bit 10 "Target reached" is set in the Status word when the difference between 0x60FF *Target Velocity* and 0x606C *Velocity Actual value* is smaller than the 0x606D *Velocity window* for a longer time than 0x606E *Velocity Window Time*.
If both conditions are not met at the same time, bit 10 "Target reached" of the Status word is reset.

The value range of Object 0x606E/0 *Velocity Window Time* is 0 ... 65535 ms.

Writing to object 0x606E/0 *Velocity Window Time* automatically generates a write command to parameter *Velocity Window Time* **1277** (data set 5, all data sets in RAM only).

If object 0x606D/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Velocity Window Time* **1277**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	6E 60	00	00 00
Reply	581	4B	6E 60	00	00 00
Write Access	601	2B	6E 60	00	64 00
Reply	581	60	6E 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.22 0x606F/0 Velocity Threshold

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606F	0	Velocity Threshold	Unsigned16	rw	No	100

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x606F *Velocity Window Threshold* defines a threshold to change Bit 12 "Velocity" of the Status word in Profile Velocity mode. If the absolute value of the Actual Velocity lies for the time given over 0x6070 *Velocity Threshold Time* above the threshold 0x606F *Velocity Threshold*, the bit is reset. If the Actual Velocity falls below the defined threshold of 0x606F *Velocity Threshold*, bit 12 "Velocity" of the Status word is set.

The value range of Object 0x606F/0 *Velocity Window Threshold* is 0 ... 65535 u/s.

Writing to object 0x606F/0 *Velocity Window Threshold* automatically generates a write command to parameter *Threshold Window* **1278** (data set 5, all data sets in RAM only !).

If object 0x606F/0 *Velocity Threshold* was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Threshold Window* **1278**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	6F 60	00	00 00
Reply	581	4B	6F 60	00	64 00
Write Access	601	2B	6F 60	00	64 00
Reply	581	60	6F 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.23 0x6070/0 Velocity Threshold Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6070	0	Velocity Threshold Time	Unsigned16	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

If the absolute value of the Actual Velocity lies for the time given over 0x6070 *Velocity Threshold Time* above the threshold 0x606F *Velocity Threshold*, the bit is reset. If the Actual Velocity falls below the defined threshold of 0x606F *Velocity Threshold*, bit 12 "Velocity" of the Status word is set.

The value range of Object 0x6070/0 *Velocity Window Time* is 0 ... 65535 ms.

Writing to object 0x606F/0 *Velocity Window Threshold* automatically generates a write command to parameter *Threshold Window Time* **1279** (data set 5, all data sets in RAM only !).

If object 0x606F/0 *Velocity Threshold* was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Threshold Window Time* **1279**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	70 60	00	00 00
Reply	581	4B	70 60	00	64 00
Write Access	601	2B	70 60	00	64 00
Reply	581	60	70 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.24 0x6071/0 Target Torque

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6071	0	Target Torque	Integer16	rw	Rx	0

The value transmitted via Object 0x6071 is selectable as source Q.808 for various parameters (e.g. *FT Input buffer percentage* **1381**).
It is also available as operation mode 95 or inverted as 195 (e.g. for parameter *Reference Percentage Source* **476** in configurations with torque control *Configuration* **30** = **x30**).

A value of 0x3E8 (=1000) corresponds to rated motor torque (100.0 %).

By default, the object 0x6071 is not connected to a device function. To use the object 0x6071 at least one device function has to be linked to the object by parameterization.

The values of Object 0x6071 is limited to -3000 to 3000 (=-300.0...300.0 %).

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6071/0	Target Torque	-3000 (= 0xF448)	3000 (= 0x0BB8)

Hexadecimal value |6071	Decimal value 0x6071	Percentage of Target Torque
0x03E8	1000	100.0
0x0064	100	10.0
0x0001	1	0.1
0xFF18	-1000	-100.0
0xFF9C	-100	-10.0
0xFFFF	-1	-0.1

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	71 60	00	00 00
Reply	581	4B	71 60	00	00 00
Write Access	601	2B	71 60	00	64 00
Reply	581	60	71 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.25 0x6077/0 Torque actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6077	0	Torque actual value	Integer16	ro	Tx	

Object 0x6077 *Torque actual value* displays the torque actual value (see parameter *Torque* **224**).

A value of 0x3E8 (=1000) corresponds to rated motor torque (100.0 %). Please refer as well to Object 0x6071.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	77 60	00	00 00
Reply	581	4B	77 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.26 0x6078/0 Current actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6078	0	Torque actual value	Integer16	ro	Tx	

Object 0x6078 *Current actual value* displays the current actual value (see parameter *R.m.s current* **211**).

A value of 0x3E8 (=1000) corresponds to the rated motor current (100.0 %). The rated motor current is set during the commissioning in parameter *Rated Current* **371**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	78 60	00	00 00
Reply	581	4B	78 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.27 0x6079/0 DClink circuit voltage

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6079	0	DClink circuit voltage	Integer32	ro	Tx	

Object 0x6079 *DC link circuit voltage* displays the actual value of the DC link voltage in mV (see parameter *DC-Link Voltage* **222**).

A value of 0x0001 86A0 (=100 000) corresponds to 100.000 V (three decimal points).

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	79 60	00	00 00 00 00
Reply	581	43	79 60	00	CA E8 04 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.28 0x607A/0 Target position

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x607A	0	Target position	Integer32	rw	Rx	0

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Positioning mode** o **Cyclic Sync Position mode**	• Motion Control: o **Velocity mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x607A *target position* defines the position (in user units) that the drive should move to in profile position mode.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	7A 60	00	00 00 00 00
Reply	581	43	7A 60	00	00 00 00 00
Write Access	601	23	7A 60	00	40 E2 01 00
Reply	581	60	7A 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.29 0x607C/0 Home offset

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x607C	0	Target position	Integer32	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Homing mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Profile Velocity mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x607C *home offset* defines the offset between the zero position of the position measurement device found during homing and the zero position of the application. All subsequent movements are in relation to the application zero position.

Writing to object *home offset* automatically generates a write command to parameter *Home Offest* **1131** (data set 5, all data sets in RAM only !).

If object 0x607C/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Home Offset* **1131**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	7C 60	00	00 00 00 00
Reply	581	43	7C 60	00	00 00 00 00
Write Access	601	23	7C 60	00	80 38 01 00
Reply	581	60	7C 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.30 0x6081/0 Profile velocity [u/s]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6081	0	Profile velocity	Unsigned32	rw	Rx	0x5 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x6081 *profile velocity* is the velocity (in user units per second) at the end of the acceleration ramp in profile position mode.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The values of Object 0x6081 are limited to 1 to 0x7FFF FFFF.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6081/0	Profile velocity (u/s)	1	2147483647 (= 0x7FFF FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	81 60	00	00 00 00 00
Reply	581	43	81 60	00	00 00 05 00
Write Access	601	23	81 60	00	40 E2 01 00
Reply	581	60	81 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.31 0x6083/0 Profile acceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6083	0	Profile acceleration	Unsigned32	rw	Rx	0x5 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode** o **Profile Positioning mode** o **Interpolated mode** o **Electronic Gear: Slave**	• Motion Control: o **Velocity mode** o **Homing mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from limit switch** • **Non motion Control** (conf. ≠ x40)

Object 0x6083 *profile acceleration* is the acceleration (in user units per second^2) in profile position mode.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The values of Object 0x6083 are limited to 1 to 0x7FFF FFFF.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6083/0	Profile acceleration (u/s²)	1	2147483647 (= 0x7FFF FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	83 60	00	00 00 00 00
Reply	581	43	83 60	00	00 00 05 00
Write Access	601	23	83 60	00	40 E2 01 00
Reply	581	60	83 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.32 0x6084/0 Profile deceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6084	0	Profile deceleration	Unsigned32	rw	Rx	0x5 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode** o **Profile Positioning mode** o **Interpolated mode** o **Electronic Gear: Slave**	• Motion Control: o **Velocity mode** o **Homing mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from limit switch** • **Non motion Control** (conf. ≠ x40)

Object 0x6084 *profile deceleration* is the deceleration (in user units per second^2) in profile position mode.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The values of Object 0x6084 are limited to 1 to 0x7FFF FFFF.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6084/0	Profile deceleration (u/s²)	1	2147483647 (= 0x7FFF FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	84 60	00	00 00 00 00
Reply	581	43	84 60	00	00 00 05 00
Write Access	601	23	84 60	00	C0 D4 01 00
Reply	581	60	84 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.33 0x6085/0 Quick stop deceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6085	0	Quick stop deceleration	Unsigned32	rw	No	0xA 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Positioning mode** o **Profile Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**	• Motion Control: o **Velocity mode** • **Non motion Control** (conf. ≠ x40)

Object 0x6085 *Quick stop deceleration* is the deceleration (in user units per second^2) in profile position mode for quick stop mode (Control word bit 2 = 0).

Writing to object *Quick stop deceleration* automatically generates a write command to parameter *Emergency Ramp* **1179** (data set 5, all data sets in RAM only !).

If object 0x6085/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory. After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Emergency Ramp* **1179**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The values of Object 0x6085 are limited to 1 to 0x7FFF FFFF.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6085/0	Quick stop deceleration (u/s²)	1	2147483647 (= 0x7FFF FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	85 60	00	00 00 00 00
Reply	581	43	85 60	00	00 00 0A 00
Write Access	601	23	85 60	00	00 00 0B 00
Reply	581	60	85 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.34 0x6086/0 Motion profile type

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6086	0	Motion profile type	Integer16	rw	No	3

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Positioning mode** o **Profile Velocity mode** o **Interpolated mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**	• Motion Control: o **Velocity mode** o **Homing mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** • **Non motion Control** (conf. ≠ x40)

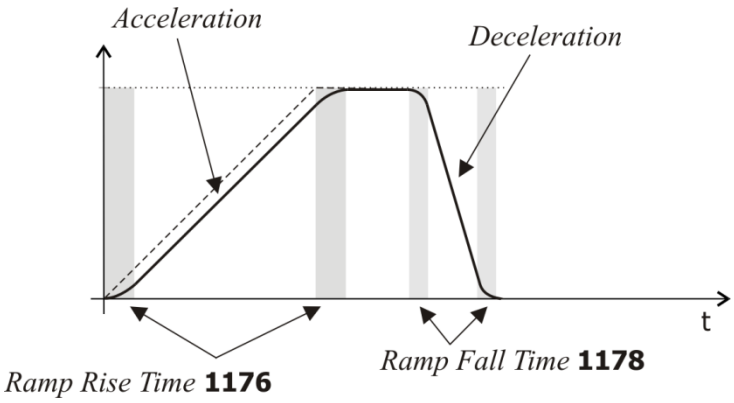
Object 0x6086 *motion profile type* defines the ramp behaviour for acceleration/deceleration.

Supported values for *motion profile type*:

0 – linear ramp
3 – jerk limited ramp

In mode 3 "jerk limited ramp", the ramp uses the parameters:

Ramp Rise Time **1176**
Ramp Fall Time **1178**



- The Ramp Rise/Fall time in *Table travel record mode* is defined via parameters **1205** and **1207**.
- The Ramp Rise/Fall time in *Homing mode* is defined via parameter **1135**.
- The Ramp Rise/Fall time in *Velocity mode* and in Non-Motion Control configurations is defined via parameters **430**...**433**.
- The Ramp Rise/Fall times in these modes are independent of the settings of object 0x6086.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	86 60	00	00 00
Reply	581	4B	86 60	00	03 00
Write Access	601	2B	86 60	00	03 00
Reply	581	60	86 60	00	00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.35 0x6091/n Gear ratio

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6091	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Motor shaft revolutions	Unsigned32	rw	No	1
	2	Driving shaft revolutions	Unsigned32	rw	No	1

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

Object 0x6091 *gear ratio* defines the ratio of motor shaft revolutions to driving shaft revolutions.

$$\text{Gear ratio} = \frac{\text{0x6091/1 } \textit{motor shaft revolutions}}{\text{0x6091/2 } \textit{driving shaft revolutions}}$$

$$\widehat{=} \frac{\text{Parameter } \textit{Gear Box : Motor Shaft Revolutions } \textbf{1117}}{\text{Parameter } \textit{Gear Box : Driving Shaft Revolutions } \textbf{1116}}$$

Writing to object *motor shaft revolutions* automatically generates a write command to parameter *Gear Box: Motor Shaft Revolutions* **1117** (data set 5, all data sets in RAM only!).

If object 0x6091/1 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of *Gear Box: Motor shaft revolutions* **1117**.

Writing to object *driving shaft revolutions* automatically generates a write command to parameter *Gear Box: Driving Shaft Revolutions* **1116** (data set 5, all data sets in RAM only!).

If object 0x6091/2 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Gear Box: Driving Shaft Revolutions* **1116**.

Alternatively the parameters **1116** and **1117** can be used instead of the Objects.

Object	Parameter
0x6091/1 Motor Shaft revolutions	**1117** *Gear Box: Motor Shaft Revolutions*
0x6091/2 Driving Shaft revolutions	**1116** *Gear Box: Driving Shaft Revolutions*

The values of Object 0x6091/1 and 6091/2 are limited as follows:

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6091/1	Motor shaft revolutions	1	65535 (= 0x0000 FFFF)
0x6091/2	Driving shaft revolutions	1	65535 (= 0x0000 FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	91 60	01	00 00 00 00
Reply	581	43	91 60	01	01 00 00 00
Write Access	601	23	91 60	01	64 00 00 00
Reply	581	60	91 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.36 0x6092/n Feed constant

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6092	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Feed	Unsigned32	rw	No	0x1 0000
	2	(Driving) shaft revolutions	Unsigned32	rw	No	1

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

Object 0x6092 *feed constant* defines the feed (in user units) per driving shaft revolutions.

$$\text{Feed constant} = \frac{0x6092/1\ \textit{feed}}{0x6092/2\ \textit{driving shaft revolutions}}$$

$$\widehat{=}\ \frac{\text{Parameter } \textit{Feed Constant}\ \textbf{1115}}{1}$$

The allowed value for 0x6092/2 *driving shaft revolutions* is **1** only. Writing values other than 1 results in an SDO abort response.

Writing to object *feed* or *driving shaft revolutions* automatically generates a write command to parameter *Feed Constant* **1115** (data set 5, all data sets in RAM only!).

If object 0x6092/1 or 0x6092/2 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Feed constant* **1115**.

The values of Object 0x6092/1 and 6092/2 are limited as follows:

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6092/1	Feed	1	65535 (= 0x0000 FFFF)
0x6092/2	(Driving) shaft revolutions	1	1

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	92 60	01	00 00 00 00
Reply	581	43	92 60	01	00 00 01 00
Write Access	601	23	92 60	01	A0 8C 00 00
Reply	581	60	92 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.37 0x6098/0 Homing method

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6098	0	Homing method	Integer8	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control x40: o **Homing mode**	• Motion Control x40: o **Profile Positioning mode** o **Profile Velocity mode** o **Velocity mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x6098/0 *homing method* determines the method that will be used during homing. For a detailed description of the different homing modes see the application manual "Positioning".

Writing to object *homing method* automatically generates a write command to parameter *Homing Mode* **1130** (data set 5, all data sets in RAM only !).

If object 0x6098/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Homing Mode* **1130**.

Homing Method 0x6098/0		Function
0 -	No Homing	**Factory setting.** No homing; the current position value is not changed. The current position value is the value saved upon the last disconnection of the power supply.
1 -	Neg. Limit Switch & Ref. -Signal	Homing to negative HW limit switch with detection of encoder ref. signal.
2 -	Pos. Limit Switch & Ref. -Signal	Homing to positive HW limit switch with detection of encoder ref. signal.
3 -	Pos. Home-Sw., Ref.-Signal left of Edge	Homing to positive home switch with detection of encoder ref. signal. Home position is the first encoder ref. signal to the left of the edge of the home switch signal.
4 -	Pos. Home-Sw., Ref.-Signal right of Edge	Homing to positive home switch with detection of encoder ref. signal. Home position is the first encoder ref. signal to the right of the edge of the home switch signal.
5 -	Neg. Home-Sw., Ref.-Signal right of Edge	Homing to negative home switch with detection of encoder ref. signal. Home position is the first encoder ref. signal to the right of the edge of the home switch signal.
6 -	Neg. Home-Sw.: Ref.-Signal left of Edge	Homing to negative home switch with detection of encoder ref. signal. Home position is the first encoder ref. signal to the left of the edge of the home switch signal.
7 -	Pos. Lim.-Sw., Ref.-Sig. left of left Edge of Home-Sw.	Homing to home switch with detection of encoder ref. signal. Homing direction positive (clockwise).

Homing Method 0x6098/0		Function
8 -	Pos. Lim.-Sw., Ref.-Sig. right of left Edge of Home-Sw.	Reversal of direction of rotation when positive HW limit switch is reached. Home position is the first encoder ref. signal to the left or right of the left or right edge of the home switch signal.
9 -	Pos. Lim.-Sw., Ref.-Sig. left of right Edge of Home-Sw.	
10 -	Pos. Lim.-Sw., Ref.-Sig. right of right Edge of Home-Sw.	
11 -	Neg. Lim.-Sw., Ref.-Sig. right of right Edge of Home-Sw.	Homing to home switch with detection of encoder ref. signal. Homing direction negative (anticlockwise). Reversal of direction of rotation when negative HW limit switch is reached. Home position is the first encoder ref. signal to the left or right of the left or right edge of the home switch signal.
12 -	Neg. Lim.-Sw., Ref.-Sig. left of right Edge of Home-Sw.	
13 -	Neg. Lim.-Sw., Ref.-Sig. right of left Edge of Home-Sw.	
14 -	Neg. Lim.-Sw., Ref.-Sig. left of left Edge of Home-Sw.	
17 ... 30: like 1 ... 14, but without encoder ref. signal		
17 -	Neg. Limit Switch	Homing to negative HW limit switch.
18 -	Pos. Limit Switch	Homing to positive HW limit switch.
19 -	Pos. Home-Sw., left of Edge	Homing to positive home switch. Home position is at the left of the edge of the home switch signal.
20 -	Pos. Home-Sw., right of Edge	Homing to positive home switch. Home position is at the right of the edge of the home switch signal.
21 -	Neg. Home-Sw., right of Edge	Homing to negative home switch. Home position is at the right of the edge of the home switch signal.
22 -	Neg. Home-Sw., left of Edge	Homing to negative home switch. Home position is at the left of the edge of the home switch signal.
23 -	Pos. Lim.-Sw., left of left Edge of Home-Sw.	Homing to home switch. Homing direction positive (clockwise). Reversal of direction of rotation when positive HW limit switch is reached. Home position is at the left or right of the left or right edge of the home switch signal.
24 -	Pos. Lim.-Sw., right of left Edge of Home-Sw.	
25 -	Pos. Lim.-Sw., left of right Edge of Home-Sw.	
26 -	Pos. Lim.-Sw., right of right Edge of Home-Sw.	
27 -	Neg. Lim.-Sw., right of right Edge of Home-Sw.	Homing to home switch. Homing direction negative (anticlockwise). Reversal of direction of rotation when negative HW limit switch is reached. Home position is at the left or right of the left or right edge of the home switch signal.
28 -	Neg. Lim.-Sw., left of right Edge of Home-Sw.	
29 -	Neg. Lim.-Sw., right of left Edge of Home-Sw.	
30 -	Neg. Lim.-Sw., left of left Edge of Home-Sw.	
33 -	Ref.-Signal left of act. pos.	Home position is the first encoder ref. signal in negative (operation mode 33) or positive (operation mode 34) direction.
34 -	Ref.-Signal right of act. pos.	
35 -	Current Position	Current position is home position. Home offset (Parameter *Home-Offset* **1131**) is taken over as actual position value.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	98 60	00	00
Reply	581	4F	98 60	00	00
Write Access	601	2F	98 60	00	23
Reply	581	60	98 60	00	00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.38 0x6099/n Homing speeds

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6099	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	speed during search for switch	Unsigned32	rw	No	0x5 0000
	2	speed during search for zero	Unsigned32	rw	No	0x2 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Homing mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** [1)]	• Motion Control x40: o **Profile Positioning mode** o **Profile Velocity mode** o **Velocity mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** • **Non motion Control** (conf. ≠ x40)

1) Electronic Gear: Slave uses this object for the Master/Slave Position Correction Function, see chapter 14.4.10.1 "Master/Slave Position Correction".

Object 0x6099/1 *speed during search for switch* defines the speed (in user units per second) during search for switch.

Object 0x6099/2 *speed during search for zero* defines the speed (in user units per second) during search for zero. This speed is also used as reference value in the "*Move away from Limit Switch*" mode.

The values of Object 0x6099/1 and 6099/2 are limited as follows:

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6099/1	speed during search for switch	1	2147483647 (= 0x7FFF FFFF)
0x6099/2	speed during search for zero	1	2147483647 (= 0x7FFF FFFF)

Writing to object *speed during search for switch* automatically generates a write command to parameter *Fast Speed* **1132** into RAM (data set 5, all data sets in RAM only!).

If object 0x6099/1 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Fast speed* **1133**.

Writing to object *speed during search for zero* automatically generates a write command to parameter *Creep speed* **1133** into RAM (data set 5, all data sets in RAM only!).

If object 0x6099/2 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Creep speed* **1133**.

The dimension of the user units is set via objects 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	99 60	01	00 00 00 00
Reply	581	43	99 60	01	00 00 05 00
Write Access	601	23	99 60	01	B0 AD 01 00
Reply	581	60	99 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.39 0x609A/0 Homing acceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x609A	0	Homing acceleration	Unsigned32	rw	No	0x5 0000

Object works in:	Object doesn't work in:
• Motion Control: o **Homing mode** o **Move away from Limit Switch** o **Electronic Gear: Slave**	• Motion Control: o **Profile Positioning mode** o **Profile Velocity mode** o **Velocity mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** • **Non motion Control** (conf. ≠ x40)

1) Electronic Gear: Slave uses this object for the Master/Slave Position Correction Function, see chapter 14.4.10.1 "Master/Slave Position Correction".

Object 0x609A/0 *homing acceleration* defines acceleration and deceleration (in user units per second^2) during homing.
The set value is also used as reference acceleration and deceleration value in "Move away from Limit Switch" mode.

Writing to object *homing acceleration* automatically generates a write command to parameter *Acceleration* **1134** (data set 5, all data sets in RAM only !).

If object 0x609A/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Acceleration* **1134**.

The dimension of the user units is set via objects 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The values of Object 0x609A/0 are limited as follows:

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x609A/0	Homing acceleration	1	2147483647 (= 0x7FFF FFFF)

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	9A 60	00	00 00 00 00
Reply	581	43	9A 60	00	00 00 05 00
Write Access	601	23	9A 60	00	90 5F 01 00
Reply	581	60	9A 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.40 0x60C1/1 Interpolation data record

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60C1	0	Highest sub-index supported	Unsigned8	ro	No	1
	1	Interpolation data record 1	Integer32	rw	Rx	0

Object works in:	Object doesn't work in:
• Motion Control: o **Interpolated mode**	• Motion Control: o **Profile Positioning mode** o **Profile Velocity mode** o **Velocity mode** o **Homing mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x60C1/1 *interpolation data record 1* is the target position (in user units) used in interpolation position mode.

Always ensure that a valid position is stored in the Interpolated Data Record.

It is recommended to copy the actual position to the Data Record before starting the Interpolated mode.

Interpolation position mode uses synchronous RxPDOs. The last received value for object 0x60C1/1 is activated with the next SYNC.

The dimension of the user units is set via objects 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	C1 60	01	00 00 00 00
Reply	581	43	C1 60	01	00 00 05 00
Write Access	601	23	C1 60	01	18 73 01 00
Reply	581	60	C1 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.41 0x60F4/0 Following error actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60F4	0	Following error actual value	Integer32	ro	Tx	

Object works in:	Object doesn't work in:
• Motion Control: o **All modes**	• **Non motion Control** (conf. ≠ x40)

In the application manual "Positioning", the term "Contouring error" is used instead of the CANopen® term "Following error".

Object 0x60F4 shows the *following error actual value*. The value is the same like stated in parameter *Actual Contouring error* **1109**.
The allowed following error is defined by object 0x6065 *Following error window*.
The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

The Contouring error can be monitored internally to trigger a device fault if a set threshold was reached. Please refer to the Application manual "Positioning" for details concerning the parameters *Fault reaction* **1120**, *Warning threshold* **1105**, *Error Threshold* **1106** and *Contouring error Time* **1119**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	C1 60	00	00 00 00 00
Reply	581	43	C1 60	00	05 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.42 0x60F8/0 Max Slippage [u/s]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60F8	0	Max Slippage	Integer32	rw	No	0

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Cyclic Sync Velocity mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x60F8/0 *Max Slippage* can be used to trigger a warning in bit 13 "maximum slip fault" in the status word when a too high slip occurs. When the difference of stator frequency and actual speed exceeds the value set in 0x60F8 Max Slippage, Bit 13 "Max Slippage" of the Status word is set, otherwise reset.

Writing to object 0x60F8 *Max Slippage* automatically generates a write command to parameter *Max Slippage* **1275** (data set 5, all data sets in RAM only !).

If object 0x60F8/0 was written and then a save parameters command (object 0x1010) processed, the object value is stored in non volatile memory.
After the next power on of the inverter, the previously set value is reactivated and overwrites the setting of parameter *Max Slippage* **1275**.

The dimension of the user units is defined by 0x6091 *Gear ratio* and 0x6092 *Feed constant*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	F8 60	00	00 00 00 00
Reply	581	43	F8 60	00	00 00 00 00
Write Access	601	23	F8 60	00	18 73 01 00
Reply	581	60	F8 60	00	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

12.5.43 0x60FF/0 Target Velocity

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60FF	0	Target Velocity	Integer32	rw	Rx	0

Object works in:	Object doesn't work in:
• Motion Control: o **Profile Velocity mode** o **Cyclic Sync Velocity mode**	• Motion Control: o **Profile Positioning mode** o **Velocity mode** o **Homing mode** o **Interpolated mode** o **Cyclic Sync Position mode** o **Table Travel record mode** o **Move away from Limit Switch** o **Electronic Gear: Slave** • **Non motion Control** (conf. ≠ x40)

Object 0x60FF *Target Velocity* defines the reference velocity in Profile velocity mode and Cyclic Synchronous Velocity mode.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	FF 60	01	00 00 00 00
Reply	581	43	FF 60	01	00 00 00 00
Write Access	601	23	FF 60	01	18 73 01 00
Reply	581	60	FF 60	01	00 00 00 00

CB: Control byte SI: Sub Index All values in hexadecimal without leading 0x

13 Motion Control Interface (MCI)

The Motion Control Interface (MCI) is a defined interface of the ACU device for positioning control via Field bus. Typically this interface is used via a Field bus like CANopen. The Motion Control Interface allows the direct access for a Field bus to change a Position Profile, which usually consists of Target Position, Speed, Acceleration, Deceleration, Quick-Stop and mode depending additional information.
The Motion Control interface uses object 0x6060 *Modes of Operation* to change between the different modes. The supported modes according to CANopen standard DS402 are:

- 1 – Profile Position mode
- 2 – Velocity mode [rpm]
- 3 – Profile Velocity mode [u/s]
- 6 – Homing
- 7 – Interpolated mode
- 8 – Cyclic sync position mode
- 9 – Cyclic sync velocity mode

Bonfiglioli Vectron defined modes

- -1 (or 0xFF) – Table Travel record mode
- -2 (or 0xFE) – Move Away from Limit Switch
- -3 (or 0xFD) – Electronic Gear: Slave

The actual mode is displayed in 0x6061 *Modes of Operation Display*.

Changing between the Modes of Operation is possible in every operation point of the ACU.

It is recommended stopping a running operation by the PLC first, then changing 0x6060 Modes of Operation and starting again the new mode.

To use the Motion Control Interface, set **412** *Local/Remote* = „1 – Control via Statemachine". In configurations without Position control (*Configuration* **30** ≠ x40) only the velocity mode is available.

For the description of the positioning parameters please refer to the "Application manual - Positioning".

"Move away from Limit Switch mode" and "Electronic Gear: Slave mode" requires firmware 5.3.0 or newer.

"Cyclic Sync Position mode" and "Cyclic Sync Velocity mode" requires firmware 5.4.0 or newer.

13.1 Object and parameter dependencies

Depending on the object 0x6060 Modes of Operation the used objects and parameters differ. Because the different objects and parameters are used they can and must be set individually.
Using the Deceleration and Quick Stop is depends on the actual mode of operation, control commands and error reaction to communication errors (see object 0x6007/0 *abort connection option code*).

The following tables contain an overview of the different used objects and parameters. The first mentioned object or parameter mentioned in a cell is usually used. If an object relates to a parameter, this parameter is mentioned.

Parameters **1292** *Modes of Operation and following (***1293**, **1294**, **1295**, **1296** & **1297***)* are used to link internal functions to CANopen objects. Usually, these need not to be changed when using CANopen.

Mode	Homing	Velocity Mode	Profile Velocity Mode
Modes of Operation [1)2)]	**6**	**2**	**3**
Target Position			
Speed	Obj. 0x6099/1 & /2 Homing Speeds → **1132** & **1133**	**1297** *S.Target velocity*[2)] Default: 806 - Obj. 0x6042 Target Velocity	**1285** *S.Target velocity pv [u/s]* [2)] Default: 816 - Obj. 0x60FF Target Velocity
Limitation[3)]	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**
Acceleration	Obj. 0x609A/0 Acceleration → **1134**	Obj. 0x6048/0 Velocity acceleration = **420** (&422)	**1295**, *Acceleration* [2)] Default :804 - Obj. 0x6083 Profile Acceleration
Deceleration	Obj. 0x609A/0 Acceleration → **1134**	Obj. 0x6049/0 Velocity deceleration = **421** (& 423)	**1296**, *Deceleration* [2)] Default : 805 - Obj. 0x6084 Profile Deceleration
Quick Stop [4)]	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*	Obj. 0x604A/0 Velocity Quick Stop = **424** (& 425)	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*
Homing Method	Obj. 0x6098/0 Homing method →**1130**		

1) Modes of Operation is selected via **1292** *S.Modes of Operation.*
Default setting: 801 - Obj. 0x6060 Modes of Operation.
2) Parameters **1285**, **1292**, **1293**, **1294**, **1295**, **1296** & **1297** are used for the connection between CANopen Objects and internal functions. For CANopen, these do not have to be changed.
3) The limitation is always restricted by **418** *Minimum frequency* and **419** *Maximum frequency*. **1118** *Limitation* of the Position controller in configuration x40 can result in a boost above maximum frequency since the output of the Position Controller is added to the maximum frequency.
4) Quick Stop or Deceleration is used depending on Stopping behaviour **630** *Operation mode* or Communication fault reaction 0x6007/0 *abort connection option code*.

Mode	Profile Positioning mode
Modes of Operation [1)2)]	1
Target Position	**1293**, *S.Target Pos.* [2)] Default: 802 - Obj. 0x607A Target Position
Speed	**1294**, *S.Profile Vel.* [2)] Default: 803 - Obj. 0x6081 Profile Velocity
Limitation[3)]	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**
Acceleration	**1295**, *Acceleration* [2)] Default :804 - Obj. 0x6083 Profile Acceleration
Deceleration	**1296**, *Deceleration* [2)] Default : 805 - Obj. 0x6084 Profile Deceleration
Quick Stop [4)]	Obj. 0x6085/0 Quick stop deceleration → **1179** *Emergency Ramp*

1) Modes of Operation is selected via **1292** *S.Modes of Operation.*
Default setting: 801 - Obj. 0x6060 Modes of Operation.
2) Parameters **1285**, **1292**, **1293**, **1294**, **1295**, **1296** & **1297** are used for the connection between CANopen Objects and internal functions. For CANopen, these do not have to be changed. Please refer to chapter 13.6 for a description.
3) The limitation is always restricted by **418** *Minimum frequency* and **419** *Maximum frequency*. **1118** *Limitation* of the Position controller in configuration x40 can result in a boost above maximum frequency since the output of the Position Controller is added to the maximum frequency.
4) Quick Stop or Deceleration is used depending on Stopping behaviour **630** *Operation mode* or Communication fault reaction 0x6007/0 *abort connection option code*.

Mode	Interpolated position mode	Cyclic Sync Position mode	Cyclic Sync Velocity mode
Modes of Operation [1)2)]	7	8	9
Target Position	0x60C1/1 interpolation data record	**1293**, *S.Target Pos.* [2)] Default: 802 - Obj. 0x607A Target Position	
Speed			**1285** *S.Target velocity pv [u/s]* [2)] Default: 816 - Obj. 0x60FF Target Velocity
Limitation[3)]	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**
Acceleration	**1295**, *Acceleration* [2)] Default :804 - Obj. 0x6083 Profile Acceleration		
Deceleration	**1296**, *Deceleration* [2)] Default : 805 - Obj. 0x6084 Profile Deceleration		
Quick Stop [4)]	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*

1) Modes of Operation is selected via **1292** *S.Modes of Operation.*
Default setting: 801 - Obj. 0x6060 Modes of Operation.

2) Parameters **1285**, **1292**, **1293**, **1294**, **1295**, **1296** & **1297** are used for the connection between CANopen Objects and internal functions. For CANopen, these do not have to be changed. Please refer to chapter 13.6 for a description.

3) The limitation is always restricted by **418** *Minimum frequency* and **419** *Maximum frequency*. **1118** *Limitation* of the Position controller in configuration x40 can result in a boost above maximum frequency since the output of the Position Controller is added to the maximum frequency.

4) Quick Stop or Deceleration is used depending on Stopping behaviour **630** *Operation mode* or Communication fault reaction 0x6007/0 *abort connection option code*.

Mode	Table Travel Record mode	Move away from limit switches	Electronic Gear: Slave
Modes of Operation [1)2)]	**-1**	**-2**	**-3**
Target Position	**1202** *Target Position*		
Speed	**1203** *Target Speed*	Obj. 0x6099/1 & /2 Homing Speeds → **1132** & **1133**	**1285** *S.Target velocity pv [u/s]* [2)] Default: 816 - Obj. 0x60FF Target Velocity
Limitation[3)]	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**	Obj. 0x6046/1 & /2 Velocity min max amount = **418 & 419**
Acceleration	**1204** *Acceleration*	Obj. 0x609A/0 Acceleration → **1134**	**1295**, *Acceleration* [2)] Default :804 - Obj. 0x6083 Profile Acceleration
Deceleration	**1205** *Deceleration*	Obj. 0x609A/0 Acceleration → **1134**	**1296**, *Deceleration* [2)] Default : 805 - Obj. 0x6084 Profile Deceleration
Quick Stop [4)]	Obj. 0x6085/0 Quick stop deceleration → **1179** *Emergency Ramp*	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*	Obj. 0x6085/0 Quick stop deceleration →**1179** *Emergency Ramp*
Motion Block	Selected via Control Word.		
Gear factor			**1123** *Gear Factor Numerator*; 0x5F10/1 Gearfactor Numerator **1124** *Gear Factor Denominator*; 0x5F10/2 Gearfactor Denominator
Phasing [5)]			**1125** *Phasing: Offset*; 0x5F11/1 Phasing 1 Offs. **1126** *Phasing: Speed*; 0x5F11/2 Phasing 1 Speed **1127** *Phasing: Acceleration* 0x5F11/3 Phasing 1 Acceleration

1) Modes of Operation is selected via **1292** *S.Modes of Operation.*
Default setting: 801 - Obj. 0x6060 Modes of Operation.

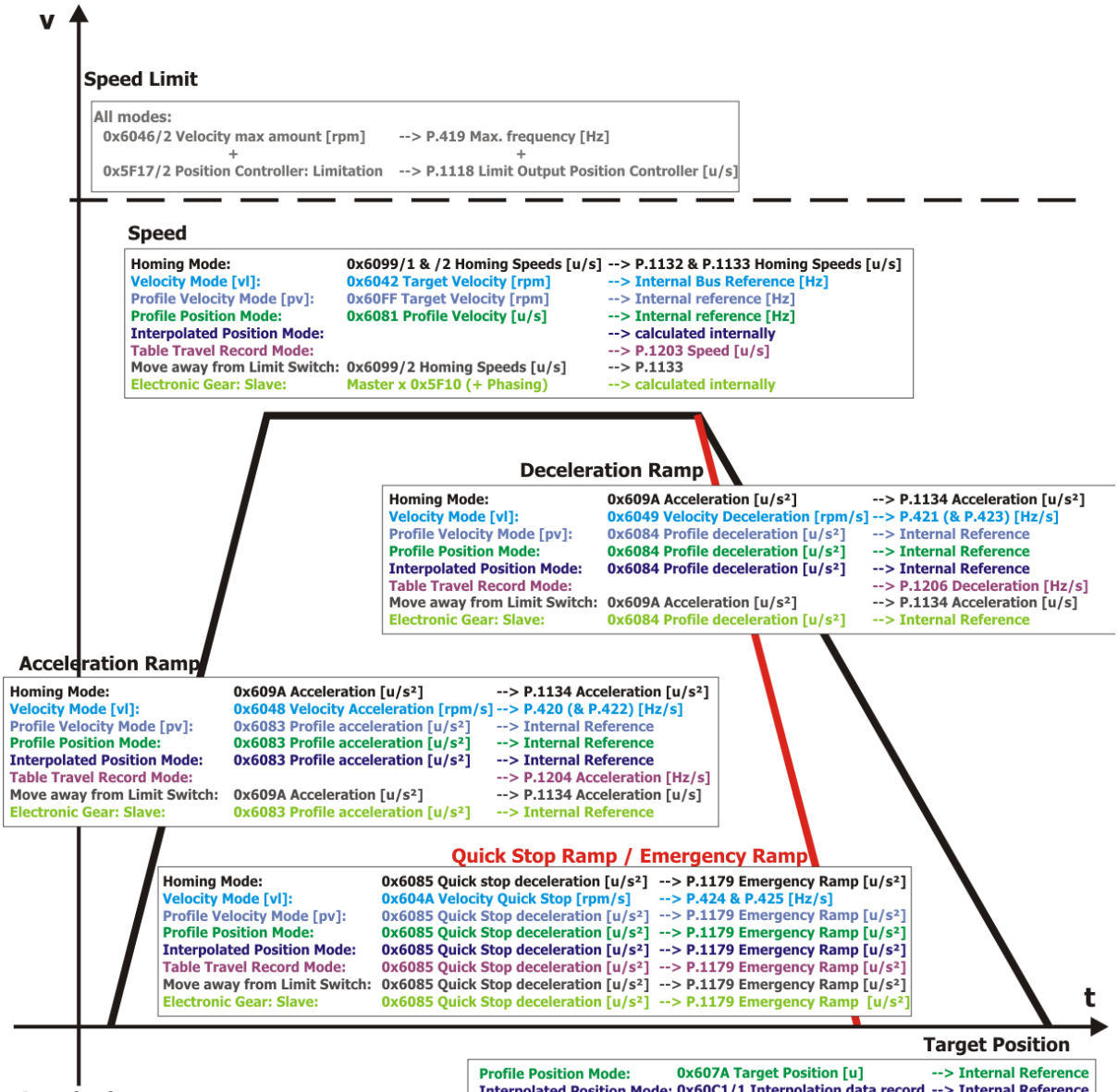
2) Parameters **1285**, **1292**, **1293**, **1294**, **1295**, **1296** & **1297** are used for the connection between CANopen Objects and internal functions. For CANopen, these do not have to be changed. Please refer to chapter 13.6 for a description.

3) The limitation is always restricted by **418** *Minimum frequency* and **419** *Maximum frequency.* **1118** *Limitation* of the Position controller in configuration x40 can result in a boost above maximum frequency since the output of the Position Controller is added to the maximum frequency.

4) Quick Stop or Deceleration is used depending on Stopping behaviour **630** *Operation mode* or Communication fault reaction 0x6007/0 *abort connection option code*.

5) Phasing is available with 4 profiles in objects 0x5F11...0x5F14.

Correlation of objects, parameters and value conversion:



Velocity [vl] → Velocity mode [rpm]
Velocity [pv] → Profile Velocity mode [u/s]

The graphical overview contains the most significant objects used. Further objects might apply to the different modes; check the descriptions of the objects and modes for further details.

The modes “Cyclic synchronous position mode” and “Cyclic synchronous velocity mode” are not shown for reasons of better readability. Please refer when using these modes to the tables and the corresponding chapters.

The Motion Control Interface (MCI) is a defined interface of the ACU device for positioning control. Typically this interface is used via a Field bus like CANopen.

13.2 Reference system

The Motion Control Interface calculates in most modes in user units. The user units result through the calculation of the gear factor and the number of pole pairs.

Conversion between user units [u] and frequencies [Hz]

$$f\,[\mathrm{Hz}] = v\left[\frac{u}{s}\right] \cdot \frac{\textit{No.of pole pairs}\ \mathbf{373} \cdot \textit{Gear Box : Driving shaft revolutions}\ \mathbf{1116}}{\textit{Feed Constant}\ \mathbf{1115}\ \frac{[u]}{U} \cdot \textit{Gear Box : Motor shaft revolutions}\ \mathbf{1117}}$$

$$v\left[\frac{u}{s}\right] = f\,[\mathrm{Hz}] \cdot \frac{\textit{Feed Constant}\ \mathbf{1115}\ \frac{[u]}{U} \cdot \textit{Gear Box : Motor shaft revolutions}\ \mathbf{1117}}{\textit{No.of pole pairs}\ \mathbf{373} \cdot \textit{Gear Box : Driving shaft revolutions}\ \mathbf{1116} \cdot}$$

Feed Constant **1115**	≙ 0x6092/1 feed
Gear Box: Motor shaft revolutions **1116**	≙ 0x6091/1 motor shaft revolutions
Gear Box: Driving shaft revolutions **1117**	≙ 0x6091/2 driving shaft revolutions

The same formulas can be used for the conversion from acceleration a [Hz/s] to a [u/s²] and vice versa. Replace in the formulas the velocities f[Hz] and v [u/s] with a [Hz/s] and a [Hz/s²].

Further details regarding the Reference system are described in the application manual “Positioning”.

13.3 Homing

When the drive is started, a defined starting position must be specified for absolute positioning modes. In a homing operation, the point of reference of the positioning operation is determined. All positioning data relates to this point of reference. Once the homing operation is started, the drive moves until it reaches a home switch or limit switch and stops there. The limit switches limit the motion path. The direction of movement (search direction) at the start of the homing operation is defined by the homing mode. Additional the reaching of a limit switch will change the direction of the drive (dependent on the homing mode). The limit switches can also be used as the point of reference.

Relative positioning and velocity operations are possible without homing.

Homing can be started:

- via a digital input
- by a control word via system bus or field bus [1)]
- automatically before the start of a motion block positioning operation

[1)] Extension module with system bus or field bus interface required

When using an Absolute Encoder with an Absolute Encoder Module (in example EM-ABS-01) a Homing after power on is not necessary. This is defined by parameter *Operation Mode* **1220**.

Further details of the Homing functions are described in the application manual “Positioning”.

13.3.1 Start position after homing

After homing:
P. 1185 = -1 → Drive remains in “coast to stop“ position
P. 1185 ≠ -1 → Drive is moved actively to set position.

13.3.2 Flying homing

The Flying homing can be used to update the reference position during a running motion. This function is described in the application manual “Positioning”.

13.4 Position Controller

The position controller evaluates the positioning operation (target/actual position) and tries to control the drive such that it comes as close as possible to the specifications.

Further details of the Position controller are explained in chapter 12.4.22 “0x5F17/n Position Controller” with a description of Object 0x5F17.

13.5 Move away from Hardware limit switches

If a hardware limit switch was activated, depending on parameter setting **1143** *Fault reaction* a fault message is triggered and the direction of rotation is disabled.

After a fault reset the still enabled direction of rotation can be used for a motion. For the moving away all operating modes can be used in general as long as the drive command executes the movement in the enabled direction.
As long as the limit switch is still triggered, the limit switch warning in the status word and in the actual parameters **269** *Warnings*, **273** *Warnings Application* und **275** *Controller status* remains set. As soon as the limit switch is free, the warning in the status word and the actual parameters is reset.

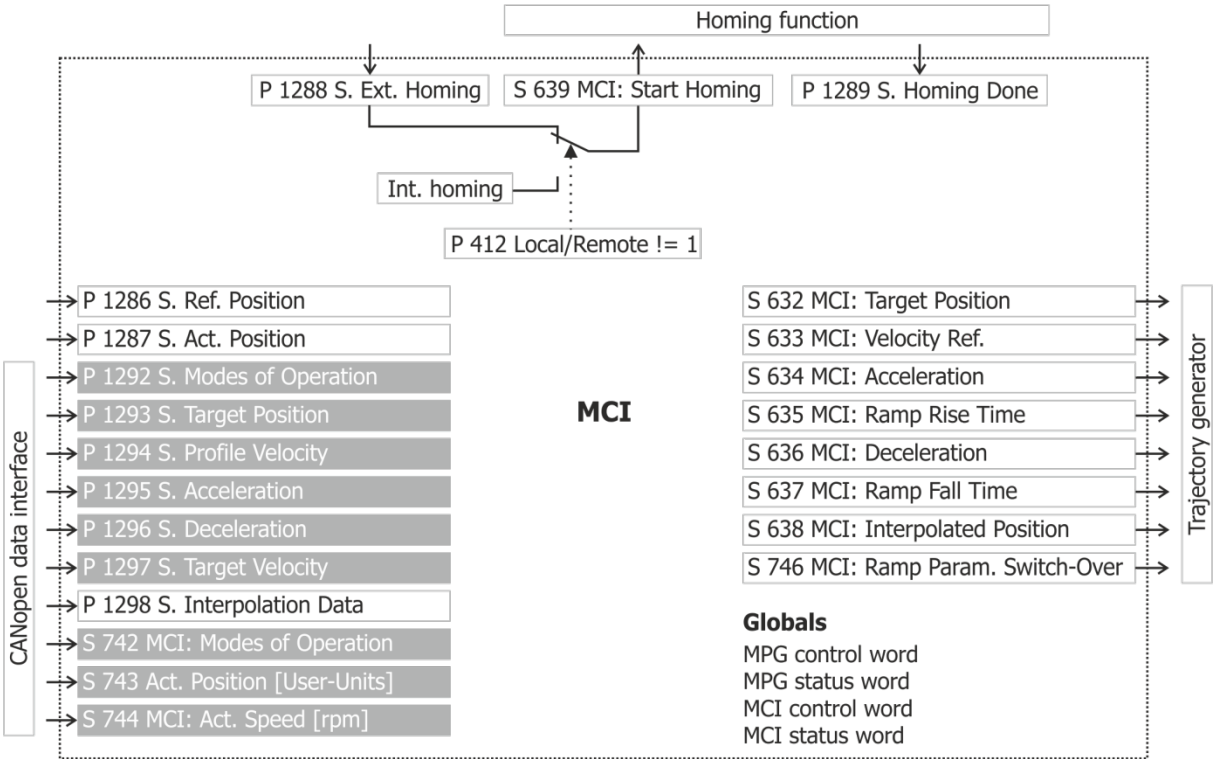
For an easy Moving away from a Limit switch the mode “-2 Move away from limit switch” can be used (chapter 14.4.9).

13.6 Motion Control Interface for Experts

The Motion Control Interface offers the experienced user the possibility to change the sources which will be used by the Motion Control Interface. By default, these are set to CANopen®. Experienced users can change these in example to Systembus sources.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
1292	S. Modes of Operation	Selection		801 – Obj. 0x6060 Modes of Operation
1293	S. Target Position	Selection		802 – Obj. 0x607A Target Position
1294	S. Profile Velocity [1)]	Selection		803 – Obj. 0x6081 Profile Velocity
1295	S. Acceleration [2)]	Selection		804 – Obj. 0x6083 Profile Acceleration
1296	S. Deceleration [3)]	Selection		805 – Obj. 0x6084 Profile Deceleration
1297	S. Target Velocity [rpm]	Selection		806 – Obj. 0x6042 Target Velocity
1299	S. Special Function Generator	Selection		9-Zero
1285	S. Target Velocity pv [u/s]	Selection		806 – Obj. 0x6042 Target Velocity

The figure below shows the parameters (P) and sources (S) that are defined in the Motion Control Interface. For CANopen®, the settings have not to be changed. The source outputs are linked to the Trajectory generator by default and also need not be changed in standard applications.



13.7 Motion Control Override

The Motion Control Override Function can be used to transmit the motion profile via serial communication (VABus or Modbus). Therefore in the User software VPlus for Windows a motion profile can be used when a control is not finished programming during commissioning. This function can also be used as a simulation mode.

The Function Motion Control Override do not support the following modes:

- Interpolated Mode.
- Cyclic Synchronous Position Mode
- Cyclic Synchronous Velocity Mode

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
1454	Override Modes of Operation	Selection		0
1455	Override Target Position	-2^{31}-1...2^{31}-1 u		-1 u
1456	Override Profile Velocity	-1...2^{31}-1 u/s		-1 u/s
1457	Override Acceleration	-1...2^{31}-1 u/s²		-1 u/s²
1458	Override Deceleration	-1...2^{31}-1 u/s²		-1 u/s²
1459	Override Target Velocity vl [rpm]	-32768...32767 rpm		-1 rpm
1460	Override Target Velocity pv [u/s]	-2^{31}-1...2^{31}-1 u/s		-1 u/s

Based on the standard settings of the Motion Control Interface (Parameters **1292**...**1297**) the following cross reference results between Override Parameters and CANopen Objects:

1454 *Override Modes of Operation*	**or**	0x6060 Modes of Operation
1455 *Override Target Position*	**or**	0x607A Target Position
1456 *Override Profile Velocity*	**or**	0x6081 Profile Velocity
1457 *Override Acceleration*	**or**	0x6083 Profile Acceleration
1458 *Override Deceleration*	**or**	0x6084 Profile Deceleration
1459 *Override Target Velocity vl [rpm]*	**or**	0x6042 Target Velocity
1460 *Override Target Velocity pv [u/s]*	**or**	0x60FF Target Velocity

The default setting „-1“ in Parameters 1455...1460 and „0“ in Parameter **1454** *Override Modes of Operation* results, that the value of the Motion Control Interface links of Parameters 1292...1297 are used. Settings deviating from the factory setting will result in the usage of the individual parameter. It is possible to set certain aspects of the trajectory via the Override function and other values via the Motion Control Interface.

The target position „-1 u“cannot be used as target position, because **1455** *Override Target Position* = -1 deactivates the Override Function.

14 Inverter Control

The control of the frequency inverter can, in principle, be carried out using three operation modes. These are set via the data set change-over capable parameter *Local/Remote* **412**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
412	Local/Remote	0	44	44

For operation with CANopen®, only the settings 0, 1 and 2 are relevant. The remaining settings relate to the possibilities of control via the KP500 control unit.

Control mode	Function
0 - Control via digital inputs (chapter 14.1)	The Start and Stop command as well as the statement of the direction of rotation are via digital input signals.
1 - Control via state machine (chapters 14.2, 14.3, 14.4)	The inverter is controlled by the *Control word*. **Only in this control mode are the motion control functions supported by *Control word* and *modes of operation* as defined with CANopen® DS402.**
2 - Control via remote digital inputs (chapter 14.1)	The Start and Stop command as well as the statement of the direction of rotation are via digital input signals emulated by the bits of the *Control word*.

Parameter *Local/Remote* **412** is data set change-over capable. Thus, it is possible to switch over between the various control modes via the data set selection.

The data set change-over can be carried out locally on the frequency inverter via digital inputs or via the bus. For data set change-over via the bus, parameter *Data set selection* **414** is used.

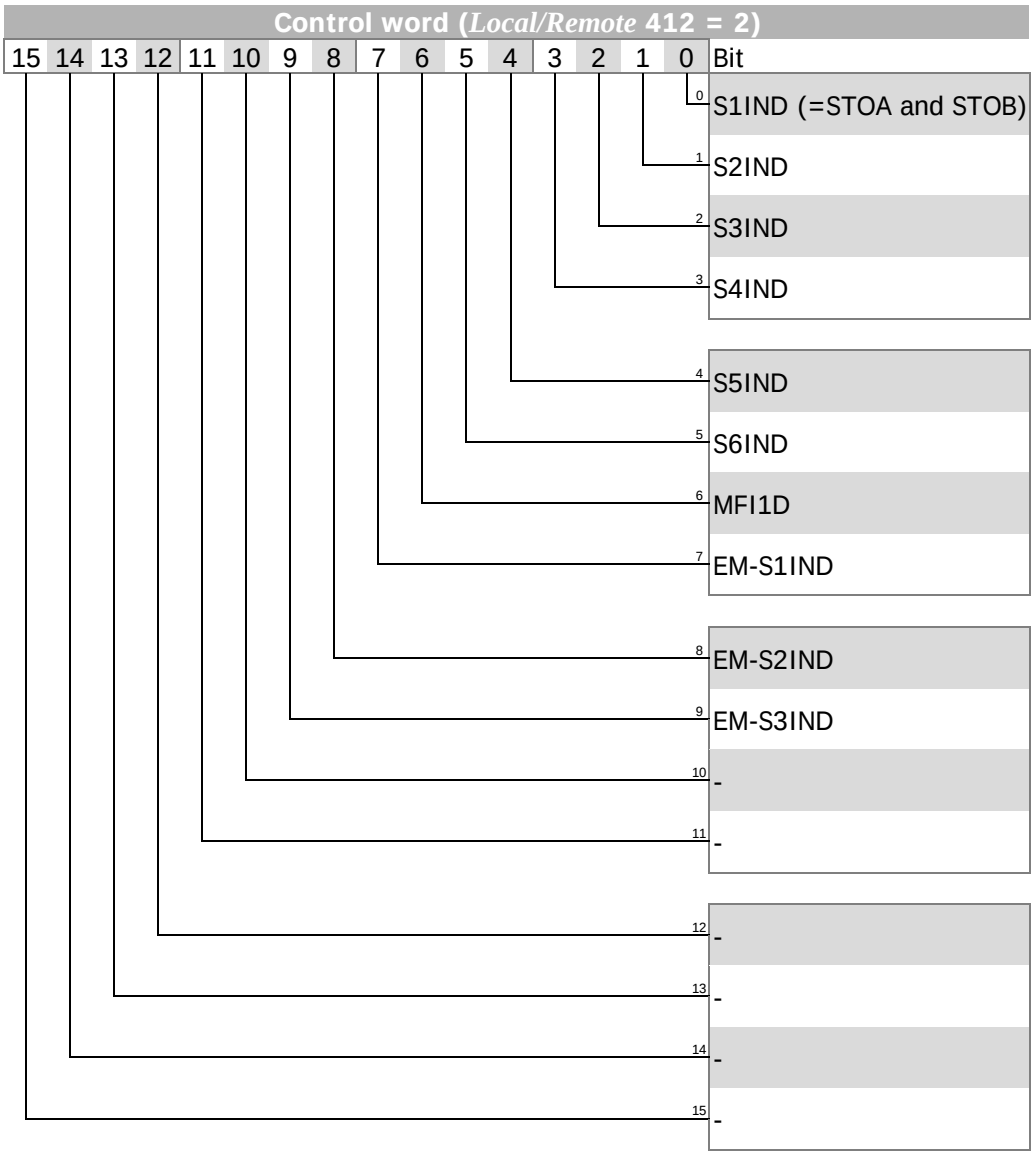
Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
414	Data set selection	0	4	0

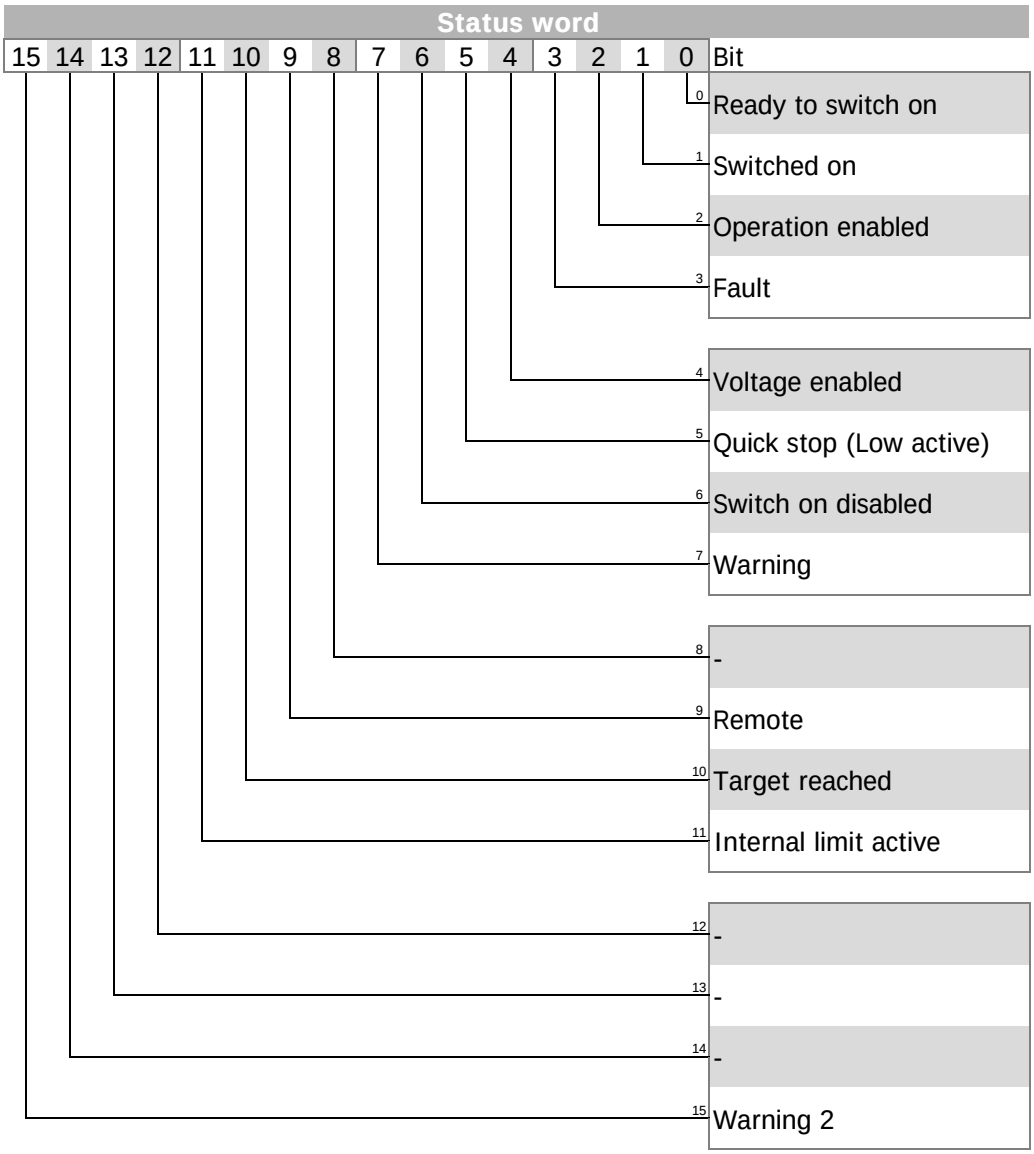
With *Data set selection* **414** = 0, the data set change-over via digital inputs is active. If *Data set selection* **414** has been set to 1, 2, 3, or 4, the corresponding data set is activated. Data set change-over via the digital inputs is then disabled.

Via parameter *Active data set* **249,** the currently selected data set can be read out. *Active data set* **249** states the activated data set with the value 1, 2, 3 or 4. This is independent of whether the data set change-over was carried out via digital inputs or via *Data set selection* **414**.

14.1 Control via digital inputs / remote digital inputs

In the operation mode Control via Contacts or Control via Remote-Contacts (parameter *Local/Remote* **412** = 0 or 2), the frequency inverter is controlled via the digital inputs S1IND (STOA AND STOB), S2IND to EM-S3IND directly, or by digital input emulation with help of the individual bits in the *Control word*. The meaning of these inputs can be taken from the operating instructions.





When using the control mode control via remote digital inputs, the digital inputs STOA AND STOB at X210A.3/X210B.2 must be set AND bit 0 of *Control word* must be set in order to get the drive started.

The two control modes, digital inputs and remote digital inputs, are only relevant to the *mode of operation* “velocity mode”.

ACTIVE CUBE inverters support an external 24V supply for control logic. Even if the mains are not switched on, communication between the PLC and the inverter can still be established.

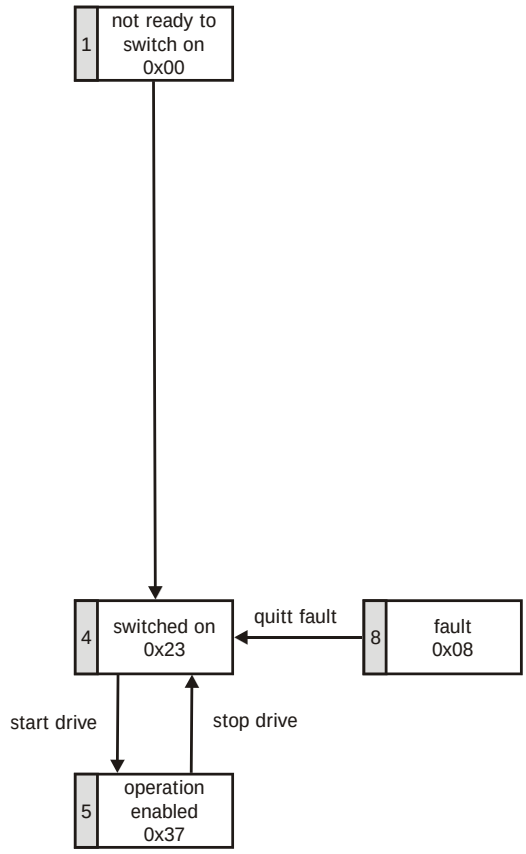
Bit 4 “Voltage enabled” of the *Status word* displays the current state of the mains power supply.

Bit 4 “Voltage enabled” = 0 signals “no mains voltage” and drive start is disabled.

Bit 4 “Voltage enabled” = 1” signals “mains voltage switched on” and drive start is enabled.

14.1.1 Device State machine

State machine:



Status word	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Switched on	1	0	0	0	1	1
Operation enabled	1	1	0	1	1	1
Fault	x	x	1	x	x	x

Bits marked “x” are don’t care.

Bit no. 7, **Warning**, can be set at any time. It indicates a device-internal warning message. The evaluation of the present warning is carried out by reading out the warning status with parameter *Warnings* **270**.

Bit no. 10, **Target reached,** is set when the specified reference value has been reached. In the special case of power failure regulation, the bit is also set if the power failure regulation has reached the frequency 0 Hz (see operating instructions). For "Reference value reached“there is a hysteresis (tolerance range), which can be set via parameter *max. Control deviation* **549** (see operating instructions).

Bit no. 11, Internal limit active, indicates that an internal limit is active. This can, for example, be the present current limit, the torque limit or the over-voltage limit. All of these limit functions lead to the reference value being quit or not reached.

Bit no. 15, **Warning 2**, indicates a warning which leads to a fault switch-off of the frequency inverter within a short period of time. This bit is set if there is a warning for motor temperature, heat sink/inside temperature, Ixt monitoring or mains phase failure.

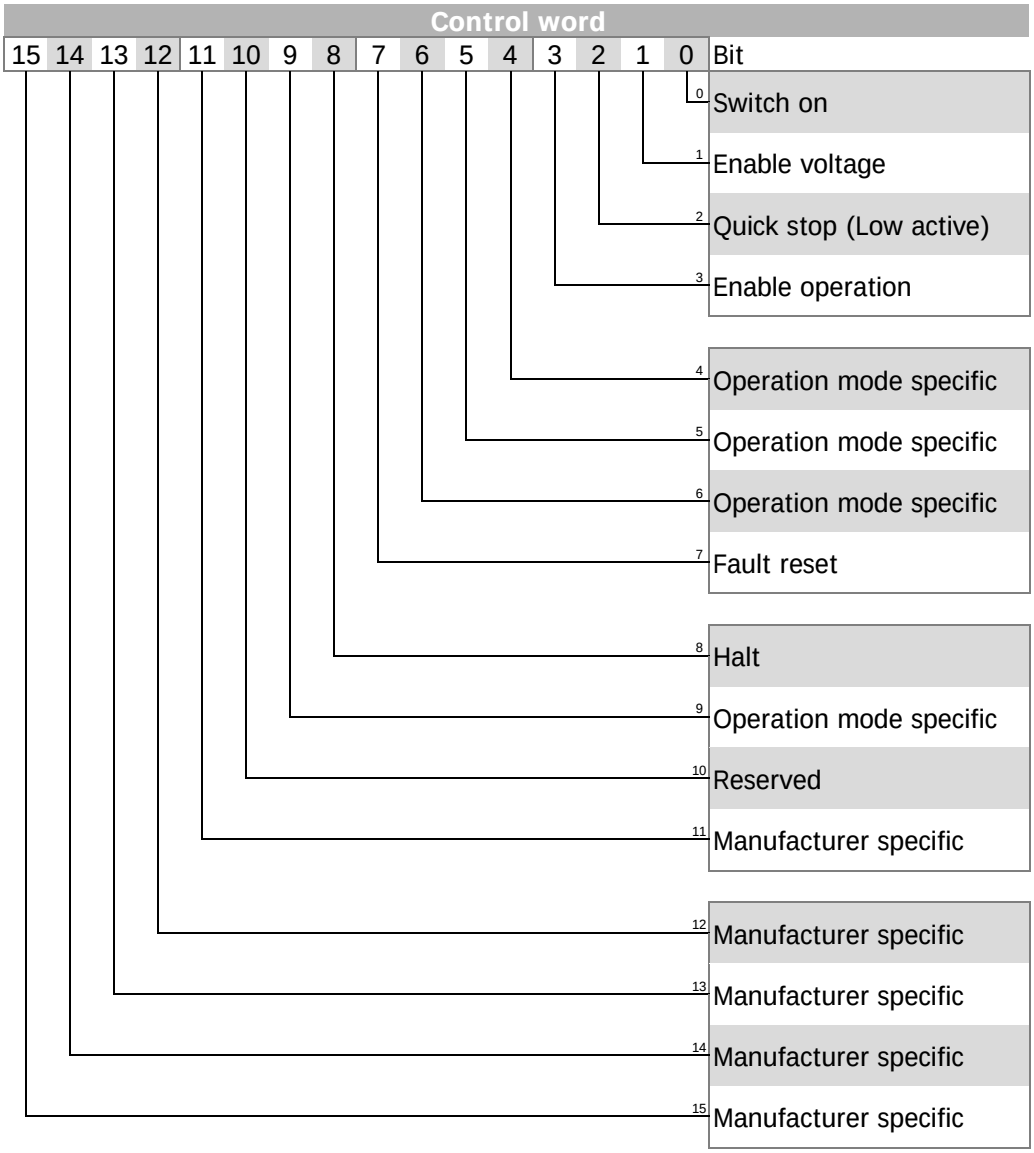
14.2 Control via state machine

In the operation mode “control via state machine” (parameter *Local/Remote* **412** = **1**), the frequency inverter is controlled via the *Control word*.

State transition 4 to state “Operation enabled” is only possible when:

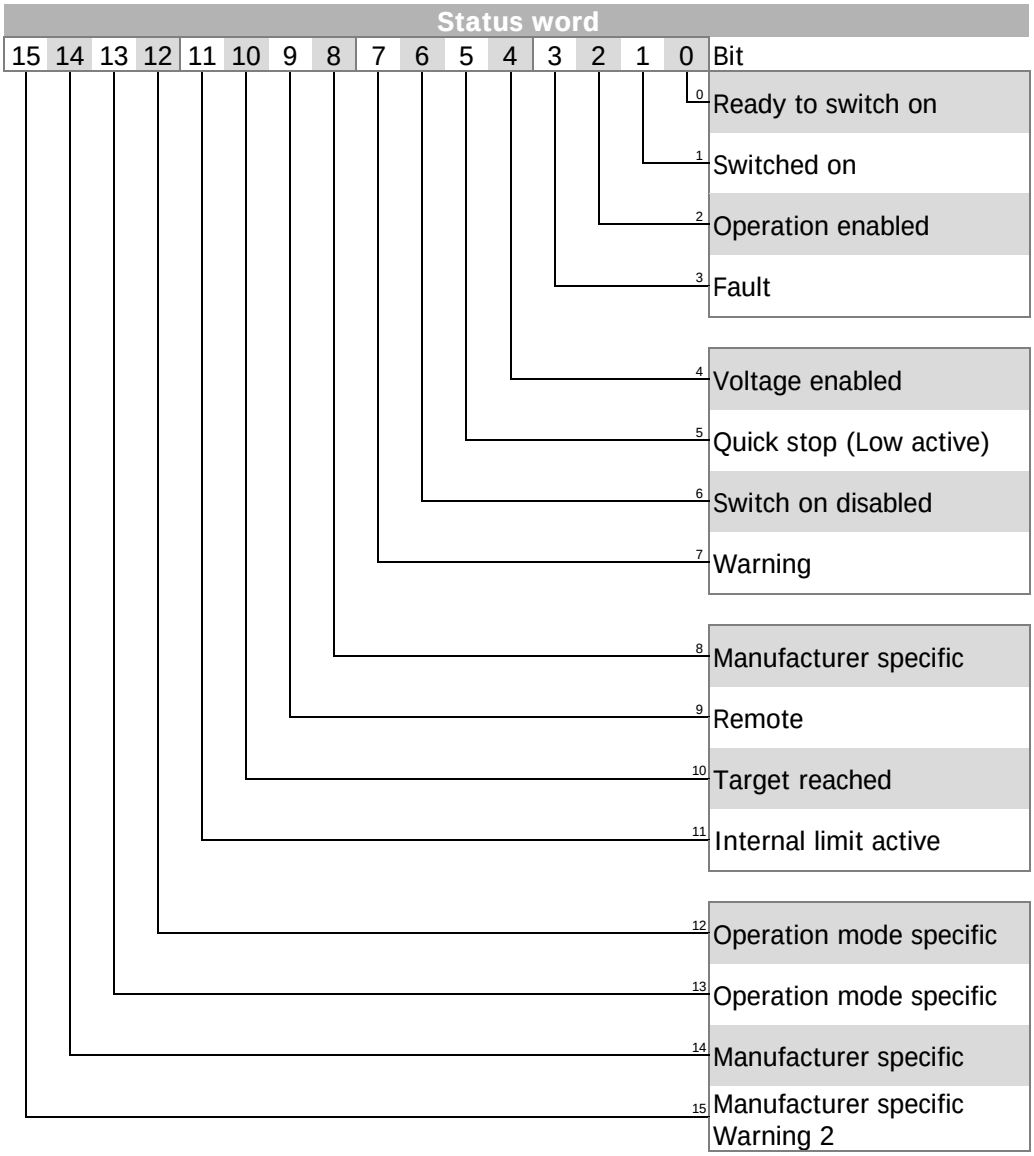
- In Motion control configuration (parameter *Configuration* **30** = x**40**) digital input S1IND (= STOA AND STOB) is set.
- In other control configurations (parameter *Configuration* **30** ≠ x**40**) digital input S1IND (= STOA AND STOB) AND (S2IND OR S3IND) is set; S2IND = start clockwise, S3IND = start anticlockwise

Object 0x6040/0 *Control word* is relevant to the inverter whenever parameter *Local/Remote* **412** is set to 1 (remote state machine).



Bits 9 ... 15 are used depending on the configuration and on 0x6060 *Mode of Operation.*

Control word bits 4, 5, 6 *operation mode specific* and bit 8 *halt* are used in motion control configurations (*Configuration* **30** = x**40**) only.



Bit 14 unused.

Status word bits 12 and 13 *operation mode specific* are used in motion control configurations (p.30 = x**40**) only.

ACTIVE CUBE inverters support an external 24V supply for control logic. Even if the mains are not switched on, communication between the PLC and the inverter can still be established.

Bit 4 "**Voltage enabled**" of the *Status word* indicates the current state of the mains power supply.

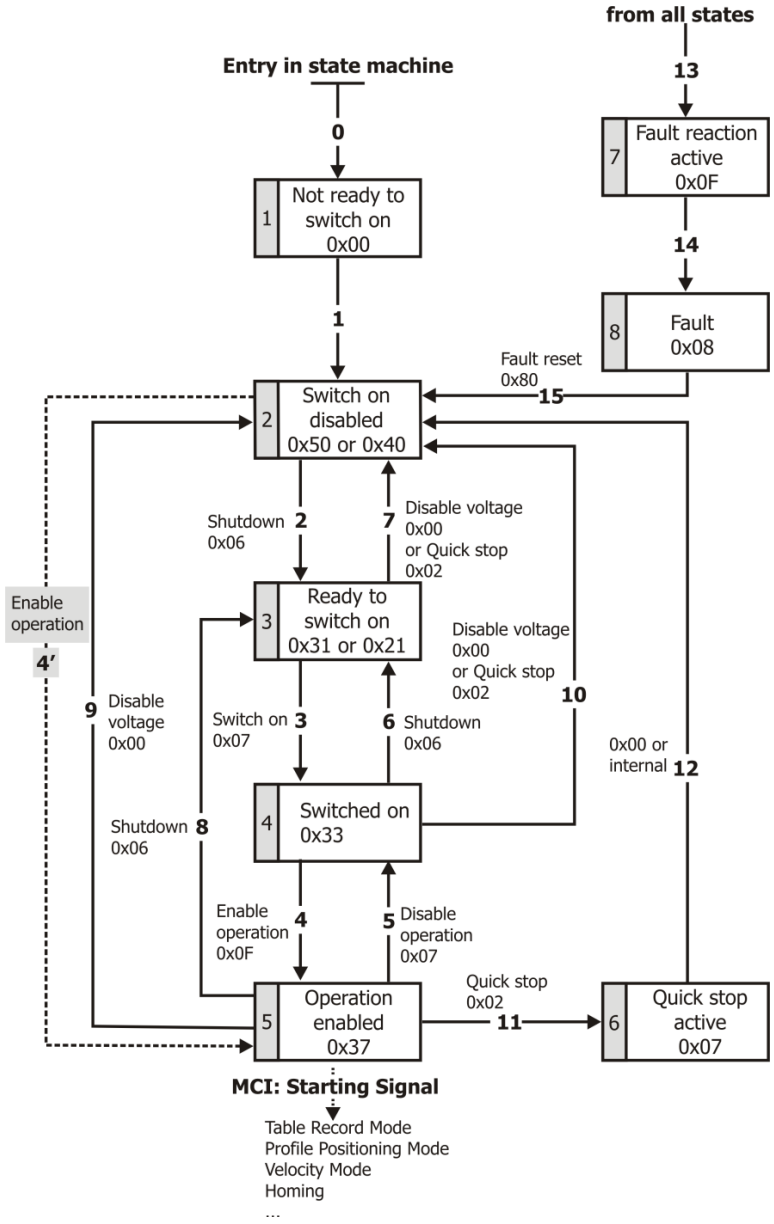
Bit 4 "**Voltage enabled**" **= 0** signals "no mains voltage" and the state transition "Ready to switch on" ➔ "Switched on" is **not possible**.

Bit 4 "**Voltage enabled**" **= 1** signals "mains voltage switched on" and the state transition "Ready to switch on" ➔ "Switched on" is **possible**.

ACTIVE CUBE inverters and ACTIVE inverters can show different states, because bit 4 of the Status word is used additionally in ACTIVE CUBE like described above.

14.2.1 Statemachine diagram

State machine:



Control word:

The device control commands are triggered by the following bit pattern in the *Control word*:

Control word

Command	Bit 7 **Fault reset**	Bit 3 **Enable operation**	Bit 2 **Quick stop**	Bit 1 **Enable voltage**	Bit 0 **Switch on**	**Transitions**
Shutdown	X	X	**1**	**1**	**0**	2, 6, 8
Switch on	X	**0**	**1**	**1**	**1**	3
Switch on	X	**1**	**1**	**1**	**1**	3
Disable voltage	X	X	X	0	X	7, 9, 10, 12
Quick stop	X	X	**0**	**1**	X	7, 10, 11
Disable operation	X	**0**	**1**	**1**	**1**	5
Enable operation	X	**1**	**1**	**1**	**1**	4
Fault reset	**0 ⇨ 1**	x	x	x	x	15

Bits marked **X** are irrelevant

State transition 3 (command “Switch on” [0x07]) is only processed if bit no. 4 “Voltage enabled” of the Status word is set.

State transition 4 (command “Enable operation” [0x0F]) is only processed if the release is set via hardware contacts STO.
If the hardware release via STO is not set, the frequency inverter remains in state “Switched On” [0x33] until the hardware release is set via STO.
If in state “Operation enable” [0x37] the STO hardware release is reset, the state is switched internally into state “Switched On” [0x33].

In configurations **with** motion control (p.30 = x**40**) consider the following points:

- State transition **4’** is **not** available.
- In status “5 – Operation enabled 0x37” an additional start signal has to be set via the “High-Byte” of the control word to start a motion of the motor. The start signal of this Motion Control Interface (MCI) is described in chapter 14.4. To change into another MCI operation mode Object 0x6060 Modes of Operation can be used.
- The controller release (STOA and STOB) must be set. Start clockwise and start anticlockwise have no function in motion control configurations.

In configurations **without** motion control (p.30 ≠ x**40**) consider the following points:

- State transition **4’** is available and is only processed if bit no. 4 “Voltage enabled” of the Status word is set. This function is for downward compatibility to older software versions.
- The inverter can only be controlled via the state machine if the logic linking is true. The logic inputs for Start clockwise / Start anticlockwise can be connected directly to ON/OFF (p.68, p.69).
  The controller release (STOA and STOB) must be set.
  Therefore this results in:
  Release: (= STOA AND STOB) **AND** (Start clockwise **OR** Start anticlockwise)

Status word:
The *Status word* displays the current operation state.

Status word

state	Bit 6 **Switch on disabled**	Bit 5 **Quick stop**	Bit 3 **Fault**	Bit 2 **Operation enabled**	Bit 1 **Switched on**	Bit 0 **Ready to switch on**
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 enabled	**0**	**1**	**0**	**1**	**1**	**1**
Quick stop active	**0**	**0**	**0**	**1**	**1**	**1**
Fault reaction active	**0**	X	**1**	**1**	**1**	**1**
Fault	**0**	X	**1**	**0**	**0**	**0**

Bits marked **X** are irrelevant

Bit 7, **Warning**, can be set at any time. It indicates a device-internal warning message. The evaluation of the warning reason is carried out by reading out the warning status with the parameter *Warnings* **270**.

Bit 9, **Remote**, is set if the operation mode "control via state machine" (*Local/Remote* **412** = 1) has been set **and** the hardware release is available.

Bit 10, Target reached, is set whenever the specified reference value has been reached.
In non motion control configurations (p.30 ≠ x**40**), target reached is related to the reference speed object 0x6042 *target velocity*. In the special case of power failure regulation, the bit is also set if the power failure regulation has reached the frequency 0 Hz (see operating instructions).
For "Target reached“there is a hysteresis (tolerance range), which can be set via parameter *max. Control deviation* **549** (see operating instructions).

Bit 11, Internal limit active, indicates that an internal limit is active. This can, for example, be the present current limit, the torque limit or the over-voltage limit. All of these limit functions lead to the reference value being quit or not reached.

Bit 15, Warning 2, indicates a warning which leads to a fault switch-off of the frequency inverter within a short period of time. This bit is set if there is a warning for motor temperature, heat sink/inside temperature, Ixt monitoring or mains phase failure.

14.3 Non motion control configurations

In non motion control configurations (Parameter *Configuration* **30** ≠ x40), object 0x6060 *modes of operation* is fixed to "2" *velocity mode*. Object 0x6061 *modes of operation display* is always "2" *velocity mode*. This cannot be changed.

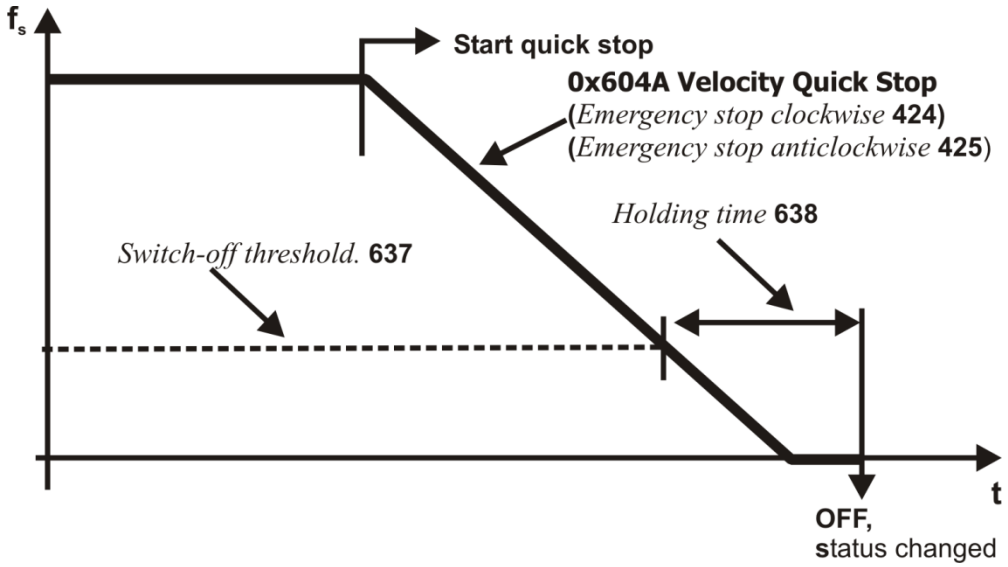
Related objects:

0x6040	Control word
0x6041	Status word
0x6042	Target velocity
0x6043	Velocity demand
0x6044	Control effort
0x6046	Velocity min max amount
0x6048	Velocity acceleration
0x6049	Velocity deceleration
0x604A	Velocity quick stop

The Ramp Rise/Fall times are set up via parameters **430**...**433**.

14.3.1 Behaviour in quick stop

In quick stop, the parameters *Switch-off threshold* **637** (percent of fmax) and *Holding time* **638** (holding time after falling short of the switch-off threshold) are relevant. In a quick stop the drive is shutdown via the emergency stop ramps. The emergency ramps are set up via Object 0x604A *Velocity Quick Stop* or parameters *Emergency stop clockwise* **424** and *Emergency stop anti-clockwise* **425**.



If frequency/speed zero has been reached during the holding time, the drive continues to be supplied with direct current until the switch-off time has expired. With this measure, there is an assurance that the drive is stationary in a change of state.

"Behaviour in quick stop" is only relevant for non motion control configurations (p.30 ≠ x**40**).

14.3.2 Behaviour in transition 5 (Disable operation)

The *behaviour in transition 5* from "Operation enabled" to "Switched on" can be parameterized. The behaviour is set via parameter *State transition 5* **392**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
392	State transition 5	0	2	2

Operation mode	Function
0 - Coast to stop	immediate transition from "Operation enabled“ to "Switched on“, free stoppage of the drive
1 - DC brake	activation of DC brake; at the end of DC braking there is a change of state from "Operation enabled“ to "Switched on“
2 - Ramp	transmission with normal stop ramp; after reaching standstill, there is a change of state from "Operation enabled“ to "Switched on“

Setting operation mode "1 - DC brake“is only possible in applications with v/f characteristic (e.g. configuration 110), as other applications do not know such an operation mode.
If the frequency inverter is operated with a configuration which does not know the DC braking operation mode (e.g. configuration 210, field-orientation speed controlled), value "1" cannot be set. It is also not offered in the selection menus of the KP500 control unit or the VPlus program.

The default value for parameter *State transition 5* **392** is operation mode "2 - Ramp". For configurations with torque control, the default value is operation mode "0 - Coast to stop". If the configuration is changed, the value set for *State transition 5* **392** is also altered, if necessary.

“Behaviour in transition 5” is only relevant for non motion control configurations (p.30 ≠ x**40**).

If transition 5 has been triggered with *State transition 5* **392** = "1 - DC brake“, a new control word is only accepted after the completion of the transition process. The change of state from "Operation enabled“ to "Ready“ is carried out after the time parameterized for the DC brake *Braking time* **632** has expired.

If the parameter *State transition 5* **392** = "2 - Ramp“has been set, the *Control word* can be set back to “Enable operation” during the stoppage of the drive. In this way, the drive runs back up to its set reference value and remains in the state "Operation enabled“.

The change of state from "Operation enabled“ to "Switched on“ is carried out after the set switch-off threshold has been reached **and** the set holding time (equivalent to the behaviour in a quick stop) has expired. In this, the parameters *Switch-off threshold* **637** (percent of fmax) and *Holding time* **638** (holding time after switch-off threshold reached) are relevant.

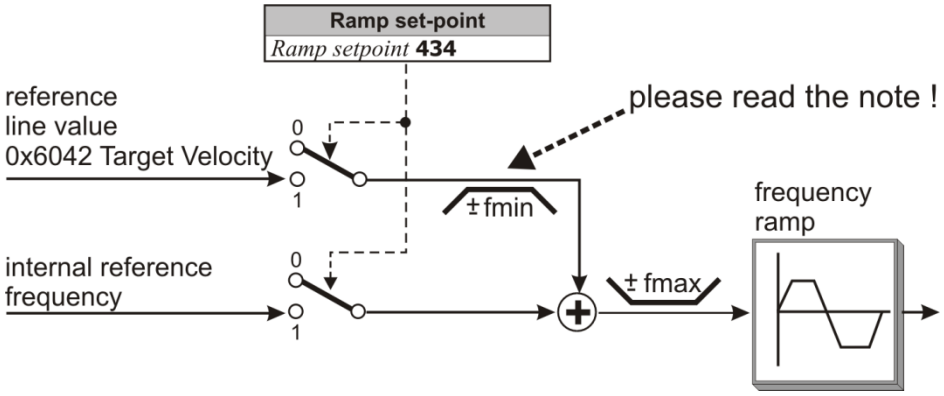
14.3.3 Reference value / actual value

The PLC gives its reference value to the frequency inverter via object 0x6042/0 *target velocity* in the RxPDO used and receives the information on its actual value back via object 0x6044/0 *control effort* in the TxPDO used.

The use of the reference/actual value channel depends on the set configuration (control system). The actual value is generated from the appropriate source depending on the control system used.

The reference value in object 0x6042/0 *target velocity* and the actual value in object 0x6044/0 *control effort* are interpreted in the notation RPM. Conversion into a frequency (reference value), or from a frequency (actual value) is carried out in the frequency inverter.

The reference value for the frequency inverter from object 0x6042/0 *target velocity* is connected to the reference line value. This reference value is combined with the internal reference value from the reference frequency value channel in the input of the ramp function. Reference frequency value channel: see operating instructions.



The internal reference value from the reference frequency value channel and the reference line value can be fed to the ramp individually or as an added variable. Setting is carried out via the data set change-over capable parameter *Ramp setpoint* **434**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Fact. sett.**
434	Ramp setpoint	1	3	3

Operation mode	Function
1 - Internal reference frequency value	Reference value from the sources of the reference frequency value channel
2 - Reference line value	Reference value via a communication interface
3 - Internal reference frequency value + reference line value	Sum of internal reference frequency value and reference line value

This function is only relevant for non motion control configurations (Parameter *Configuration* **30** ≠ x40)

If *Ramp set-point* **434** = 2 (only reference line value), then this reference line value is limited to fmin. Please remember that the sign in front of fmin at reference value = 0 is derived from the sign in front of the last reference line value ≠ 0.
After Power On, the reference line value is limited to +fmin!

For *Ramp set-point* **434** = 3, the sign in front of the overall reference value results from the sum of the internal reference frequency value and the reference line value.

The reference values can be read out from the frequency inverter with the help of the KP500 control unit or VPlus operating software.

Actual values		
Parameter	**Contents**	**Format**
Internal reference frequency **228**	Internal reference value from the frequency reference value channel	xxx.xx Hz
Reference bus frequency **282**	Reference line value from the CANopen® bus	xxx.xx Hz
Reference ramp frequency **283**	Sum of internal + reference line value	xxx.xx Hz

14.3.4 Example Sequence

To start the drive without Position control (*configuration* **30** ≠ x40), the correct sequence has to be sent from the PLC.

1	Control word =	0x0000	Disable voltage
3	Control word =	0x0006	Shutdown
4	Control word =	0x0007	Switch On
5	Control word =	0x000F	Enable Operation

OR

1	Control word =	0x0000	Disable voltage
5	Control word =	0x000F	Enable Operation

In non motion control configurations (Parameter *Configuration* **30** ≠ x40) the second (shortened) sequence can be used because state transition **4'** is available in these configurations.

14.4 Motion control configurations

⚠ WARNING

Dangerous state due to new mode!

If 0x6060 *modes of operation* is changed during operation (control word = 0xnnnF), a dangerous state may occur in the new mode.

- Before changing 0x6060 *modes of operation*, check the status word (e.g. for state 0xnn33).

Definition Motion Control

For the full function of the Motion Control Interface, you will have to set *Local/Remote* **412** = "1-Control via state machine". In all other operation modes of parameter *Local/Remote* **412**, there are major restrictions. The descriptions in this chapter and of all objects used are based on the setting *Local/Remote* **412** = "1-Control via state machine".

Settings *Local/Remote* **412** ≠ 1 are described in the "Positioning" application manual.

The function of the state machine describes the basic operating behaviour of the frequency inverter in configurations with position control (*Configuration* **30** = x40). The objects *control word* and *status word* described above support the bits marked as operation mode specific.
These bits and bit "Target reached" has different meanings in the different position control operation modes – defined by 0x6060 *modes of operation*. The following chapters describe the application of the operation mode specific bits in the *control word* and *status word*, depending on the different position control operation modes. Default value: 0x6060 *modes of operation* = 2 – velocity mode.

Basic functions:

The state machine must be set to "operation enabled", before the position command can be issued via the operation mode specific bits of the *control word*.

Once a *mode of operation* has been set by the PLC, no commands will be excepted for this operation mode until this operation mode is displayed in the *modes of operation display* object.

The bits in the *control word* and *status word* marked as operation mode specific are only supported in configurations with position control (*Configuration* **30** = x40).

14.4.1 Velocity mode [rpm]

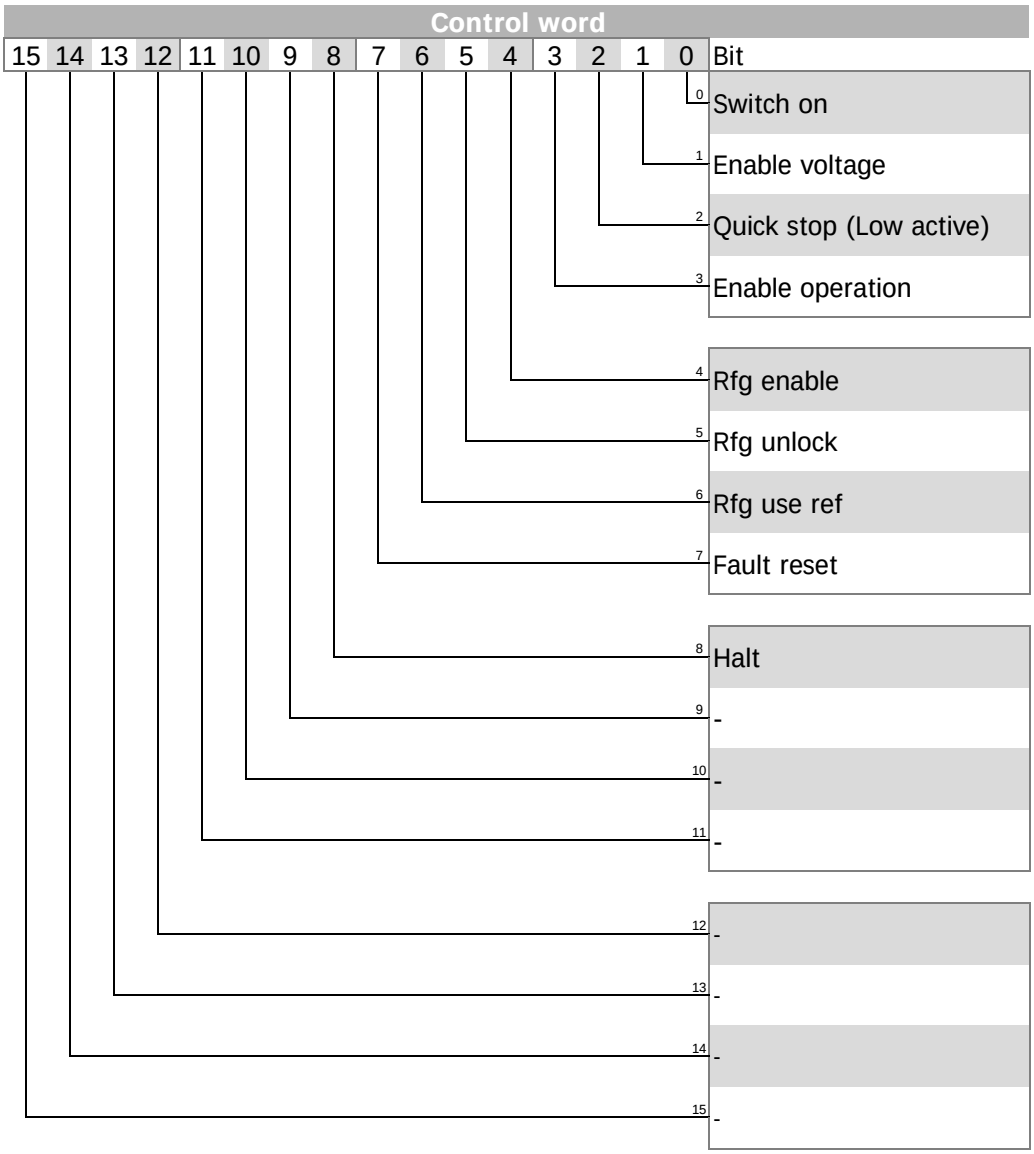
The velocity mode is selected via object 0x6060/0 *Modes of operation* = **2**.
In velocity mode the "operation mode specific" bits of the *Control word* control the ramp function generator "**rfg**". The function is explained in the block diagram.

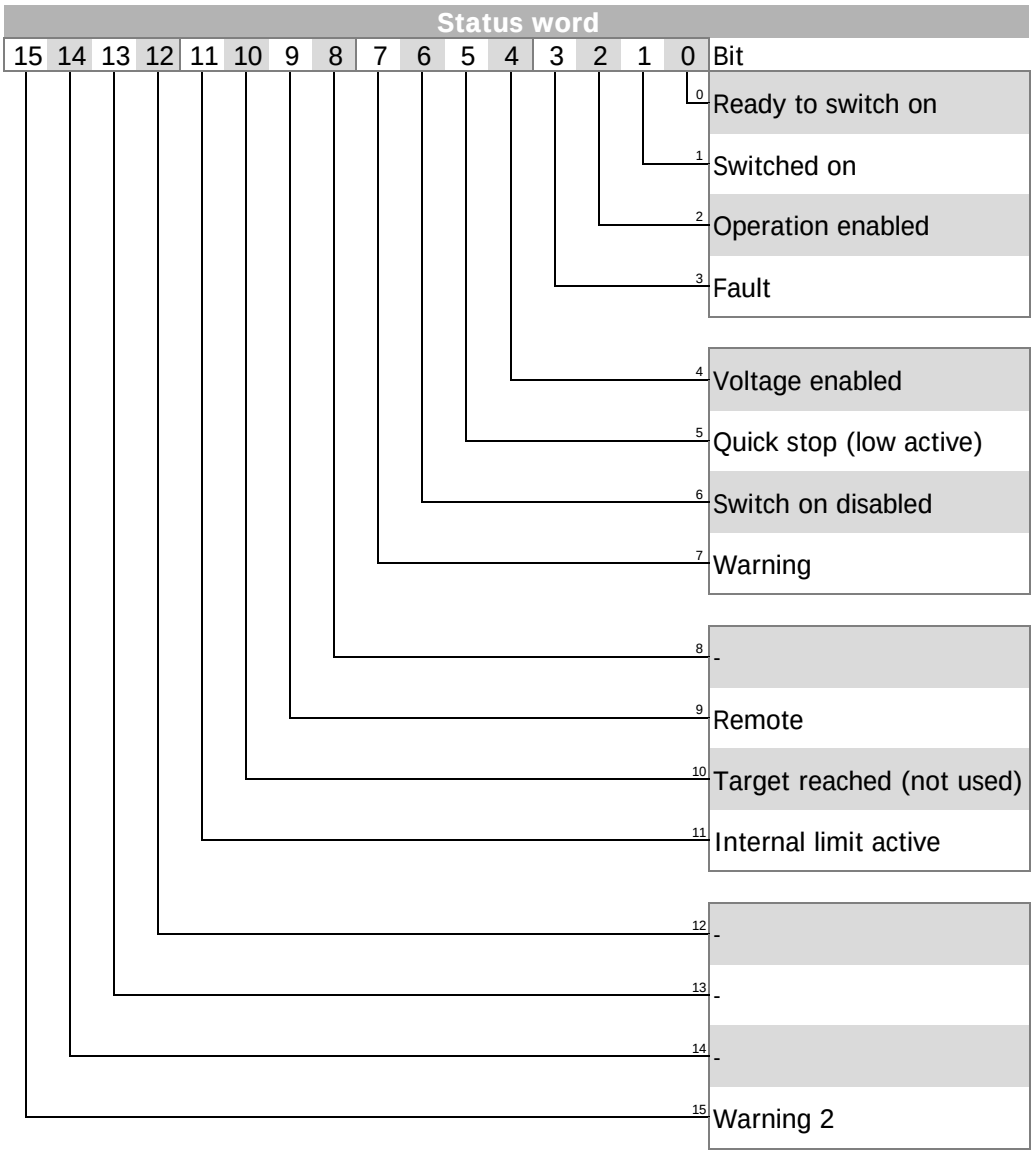
Related objects:

0x6040	Control word
0x6041	Status word
0x6042	Target velocity
0x6043	Velocity demand
0x6044	Control effort
0x6046	Velocity min max amount
0x6048	Velocity acceleration
0x6049	Velocity deceleration
0x604A	Velocity quick stop
0x6060	Modes of operation
0x6061	Modes of operation display

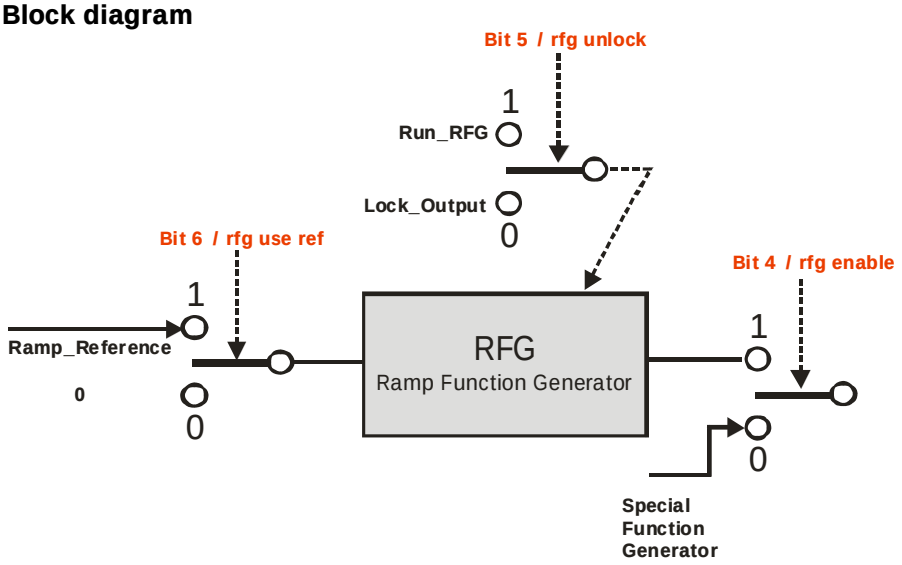
1

The Ramp Rise/Fall times are set up via parameters **430**...**433**.





Block diagram



Bit 4:rfg enable

Rfg enable = 0 The Reference speed from the manufacturer specific special function generator is used

Rfg enable = 1 The Reference speed from the ramp output is used

The special function generator is only evaluated, if **1299** *Q. Special Function Generator* is set unequal to “9 –Zero”.
If **1299** *Q. Special Function Generator* is set to “9-Zero”, the value of the ramp output is always used.

If Special function generator **1299** *Q. Special Function Generator* is set unequal to “9-Zero”, the reference value from the ramp output is also used when Bit 4 “rfg enable” = 1 and when Bit 4 “rfg enable” = 0 the reference value is used from the source set in **1299** *Q. Special Function Generator*.

Reference value source		
	1299 *Q. Special Function Generator* unequal to „9-Zero“	**1299** *Q. Special Function Generator* = „9-Zero“
Bit 4 rfg enable = 0	Reference value from Special function	Reference value from ramp output
Bit 4 rfg enable = 1	Reference value from ramp output	

Bit 5:rfg unlock
Rfg unlock = 0 The last speed value is hold and used.
Rfg unlock = 1 The ramp function is active and will change the speed according to the setpoint and the ramp.

Bit 6: rfg use ref
Rfg use ref = 0 The setpoint “0” is used.
Rfg use ref = 1 The setpoint of 0x6042 *Target Velocity* is used.

Bit 8: Halt
HALT = 0 → **execute motion**
HALT = 1 → **stop axle** (inverter remains in state “operation enabled”)

14.4.1.1 Example Sequence

To start the velocity mode, the correct sequence has to be sent from the PLC.

1	Control word = Status word =	0x0000 0x0050	Disable voltage Switch On Disabled
2	Modes of Operation =	2	(Velocity mode)
3	Control word = Status word =	0x0006 0x0031	Shutdown Ready to switch on
4	Control word = Status word =	0x0007 0x0033	Switch On Switched On
5	Control word = Status word =	0x000F 0xnn37	Enable Operation, no change to previous state if already enabled. Operation enabled
6a	Control word = Status word =	0x007F 0xnn37	Start Velocity mode with Reference speed from object 0x6042 Target velocity. Operation enabled
6b	Control word = Status word =	0x006F 0xnn37	**1299** *Q. Special Function Generator*: = "9-Zero" ➔ Starts „Velocity mode" with reference value from Object 0x6042 Target velocity. **1299** *Q. Special Function Generator*: unequal "9-Zero" Starts with reference value from source from **1299** *Q. Special Function Generator* Operation enabled
6c	Control word = Status word =	0x003F 0xnn37	Start Velocity mode with Reference speed "0". Operation enabled
6d	Control word = Status word =	0x002F 0xnn37	**1299** *Q. Special Function Generator*: = "9-Zero" ➔ Starts „Velocity mode" with reference value "0". **1299** *Q. Special Function Generator*: unequal "9-Zero" Starts with reference value from source from **1299** *Q. Special Function Generator* Operation enabled
6e	Control word = Status word =	0x005F 0xnn37	Start Velocity mode with actual speed – a ramping process is cancelled. Disable voltage
6f	Control word = Status word =	0x004F 0xnn37	**1299** *Q. Special Function Generator*: = "9-Zero" ➔ Starts „Velocity mode"with actual speed – a running ramp is interrupted. **1299** *Q. Special Function Generator*: unequal "9-Zero" Starts with reference value from source from **1299** *Q. Special Function Generator* Operation enabled
7	Control word = Status word =	0x01xx 0xnn37	HALT: The drive is stopped with ramp 0x6049 *Velocity deceleration*. Operation enabled

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

14.4.2 Profile Velocity mode [u/s]

The profile velocity mode is selected via object 0x6060/0 *Modes of operation* =**3**.
In profile Velocity mode the inverter receives a reference speed in [u/s].

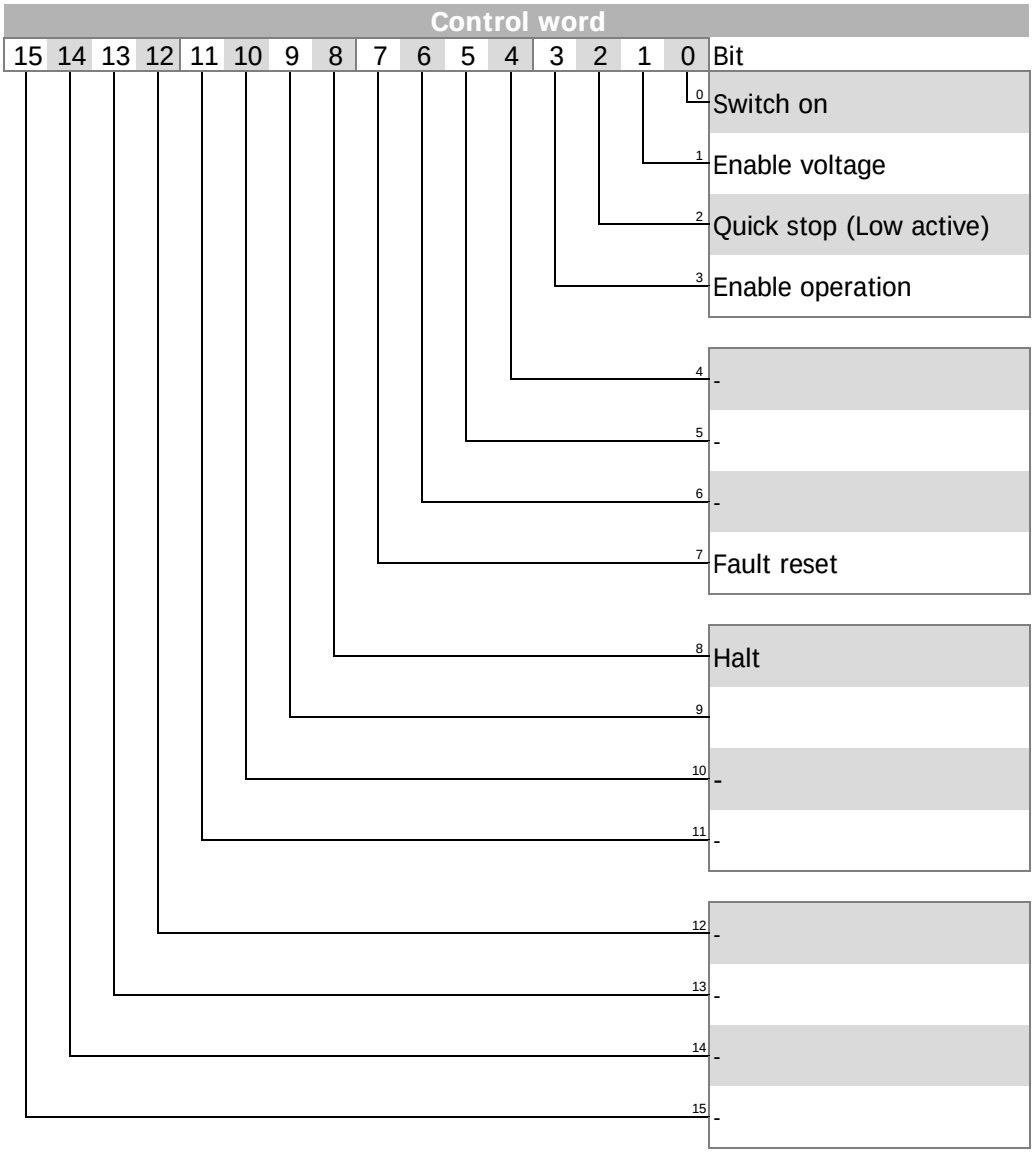
Related objects:

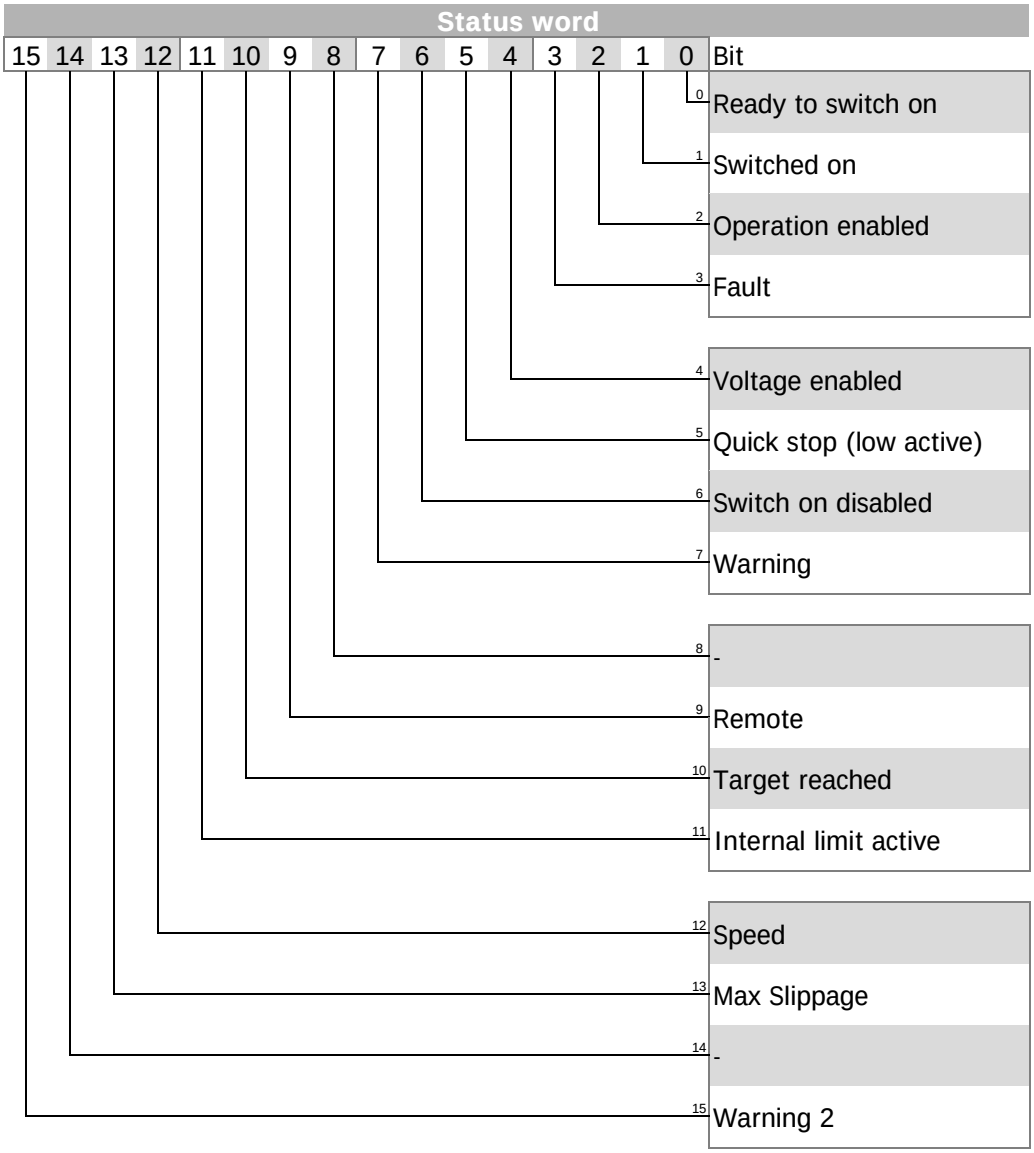
0x6040	Control word
0x6041	Status word
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x606C	Velocity Actual value
0x606D	Velocity Window
0x606E	Velocity Window Time

0x606F	Velocity Threshold
0x6070	Velocity Threshold Time
0x6083	Profile acceleration
0x6084	Profile deceleration
0x6085	Quick stop deceleration
0x6086	Motion Profile Type
0x60F8	Max Slippage
0x60FF	Target Velocity

The Ramp Rise/Fall times are set up via parameters **1176** and **1178** and object 0x6086.

In profile position mode the "operation mode specific" bits of Control word and Status word are used as shown:





The Profile Velocity Mode is used to set the reference speed in user units [u/s]. The reference speed 0x60FF *Target Velocity* is taken over in mode “operation enabled” immediately (0xnn37). The acceleration and deceleration ramp are specified by objects 0x6083 *Profile acceleration* and 0x6084 *Profile deceleration*.
Setting Bit 8 “Halt” of the control word delays the drive with ramp 0x6084 *Profile deceleration* and holds the drive at standstill. Resetting Bit 8 results in an acceleration with ramp 0x6083 *Profile acceleration* to the actual reference velocity.

Control word Bit 8: Halt
HALT = 0 → **Execute Profile Velocity Mode.**
HALT = 1 → **Halt Axis.** (The Frequency inverter remains in state “Operation enabled”.)

The actual velocity in [u/s] can be displayed in a PLC via map able Object 0x606D.

Via Objects 0x606D *Velocity Window* and 0x606E *Velocity Window time* Bit 10 “Target reached” of the status word is set.
Via Objects 0x606F *Velocity Threshold* and 0x6070 *Velocity Threshold time* Bit 12 „Velocity“of the status word is set.
Via Object 0x60F8 *Max Slippage* a slip monitoring via Bit 13 “Max Slippage” of the status word can be set up.

Status word Bit 10: Target reached

Target reached = 0 → The actual velocity doesn’t match the reference velocity.

Target reached = 1 → **The actual velocity matches the reference velocity.**

The actual velocity differs at least from the defined time period in object 0x606E *Velocity Window time* up to the defined amount [us] in Object 0x606D *Velocity Window.*

Status word Bit 12: Velocity

Velocity = 0	**→ The Actual Velocity matches the comparison speed.** The Actual Velocity has exceeded for a defined time (Object 0x6070 *Velocity Threshold time*) a defined Velocity in user units per seconds [u/s] (Object 0x606F *Velocity Threshold)*.
Velocity = 1	→ The Actual Velocity doesn’t match the Comparison Velocity.

Status word Bit 13: Maximum Slippage

Maximum Slippage = 0	**→ The actual Slippage speed is smaller than defined.** The comparison value of the slippage speed is defined Object 0x60F8 *Max Slippage*.
Maximum Slippage = 1	**→ The actual Slippage speed is bigger than defined.** The comparison value of the slippage speed is defined Object 0x60F8 *Max Slippage*.

14.4.2.1 Example Sequence

To start the Profile Velocity mode, the correct sequence has to be sent from the PLC.

1 1	Control word = Status word =	0x0000 0x0050	Disable voltage Switch On Disabled
2	Modes of Operation =	3	(Profile Velocity mode)
3	Control word = Status word =	0x0006 0x0031	Shutdown Ready to switch on
4	Control word = Status word =	0x0007 0x0033	Switch On Switched On
5	Control word = Status word =	0x0007 ↓ 0x000F 0xnn37	Enable Operation, no change to previous state if already enabled. The Profile Velocity mode is started with reference velocity 0x60FF *Target Velocity* and Ramp profile 0x6084 *Profile acceleration* and 0x6084 *Profile deceleration*. Changes to Target Velocity and Ramps are taken over immediately. Operation enabled

1) A profile consists of the following entries. If a value is not changed, the old value will still be active.

- 0x6081 *Profile velocity*
- 0x6083 *Profile acceleration*
- 0x6084 *Profile deceleration*
- 0x60FF *Target Velocity*

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

14.4.3 Profile position mode

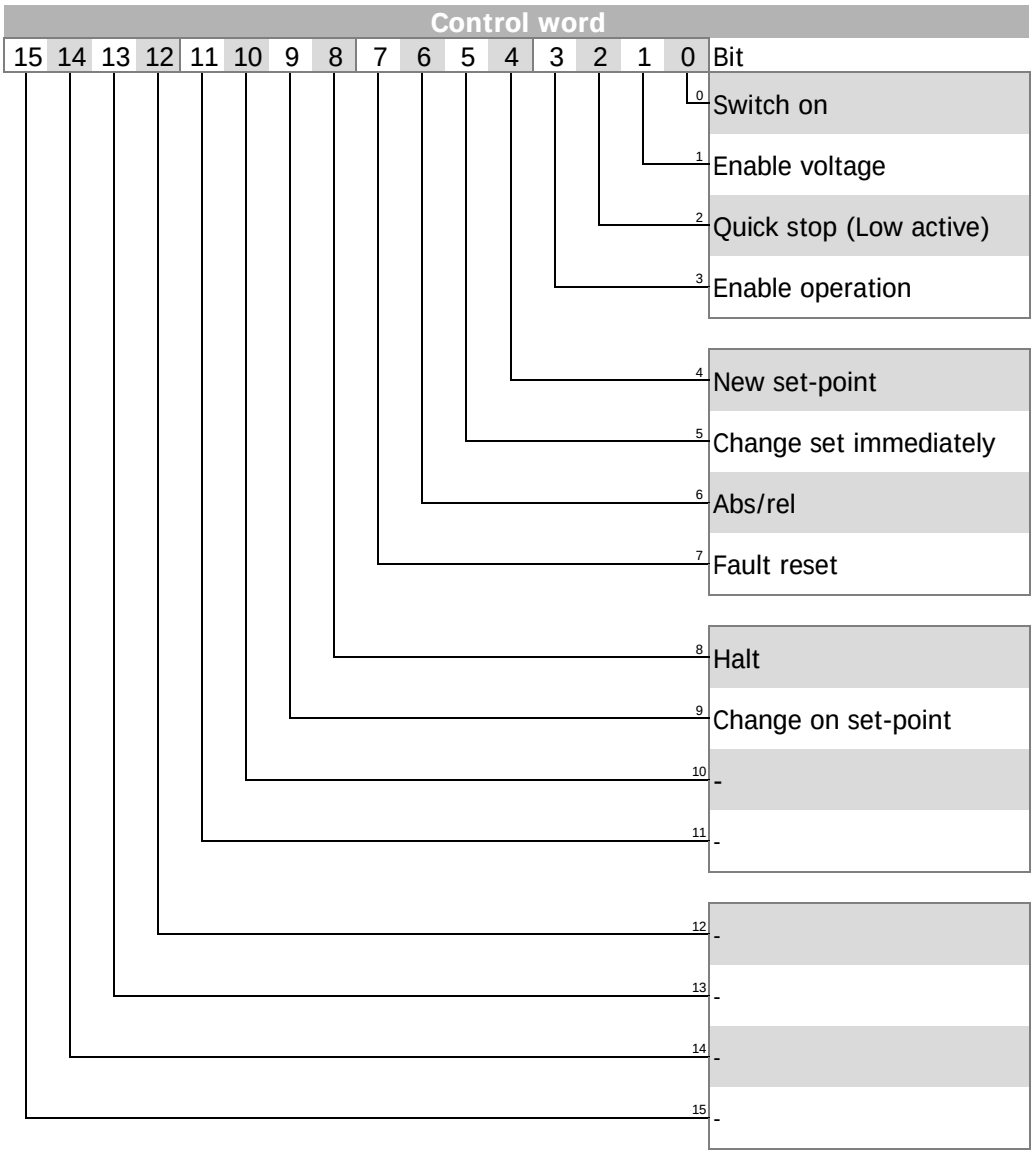
The profile position mode is selected via object 0x6060/0 *Modes of operation* =**1**.
In profile position mode the inverter receives a target position followed by a command to move to this position.

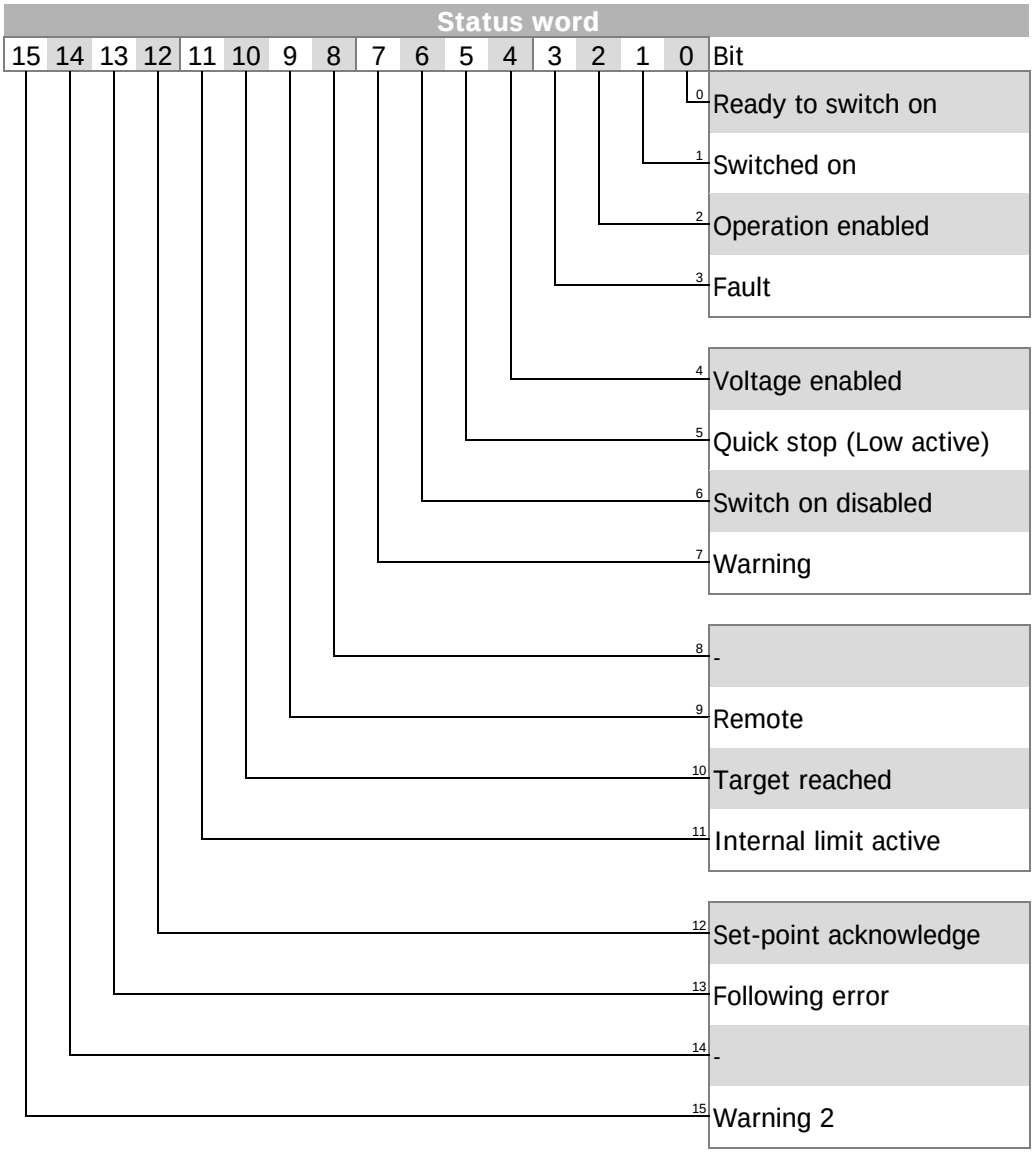
Related objects:

0x6040	Control word
0x6041	Status word
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x607A	Target position
0x6081	Profile velocity
0x6083	Profile acceleration
0x6084	Profile deceleration
0x6085	Quick stop deceleration

The Ramp Rise/Fall times are set up via parameters **1176** and **1178** and object 0x6086.

In profile position mode the “operation mode specific” bits of Control word and Status word are used as shown:




Status word
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit
0 Ready to switch on
1 Switched on
2 Operation enabled
3 Fault
4 Voltage enabled
5 Quick stop (Low active)
6 Switch on disabled
7 Warning
8 -
9 Remote
10 Target reached
11 Internal limit active
12 Set-point acknowledge
13 Following error
14 -
15 Warning 2

Control word

Change on set-point Bit 9	Change set-point immediately Bit 5	New set-point Bit 4	Description
0	0	0 → 1	Positioning shall be completed (target reached) before the next one is started
X	1	0 → 1	Next position shall be started immediately
1	0	0 → 1	Positioning with the current profile velocity up to the current set-point shall be proceeded and then the next positioning shall be applied

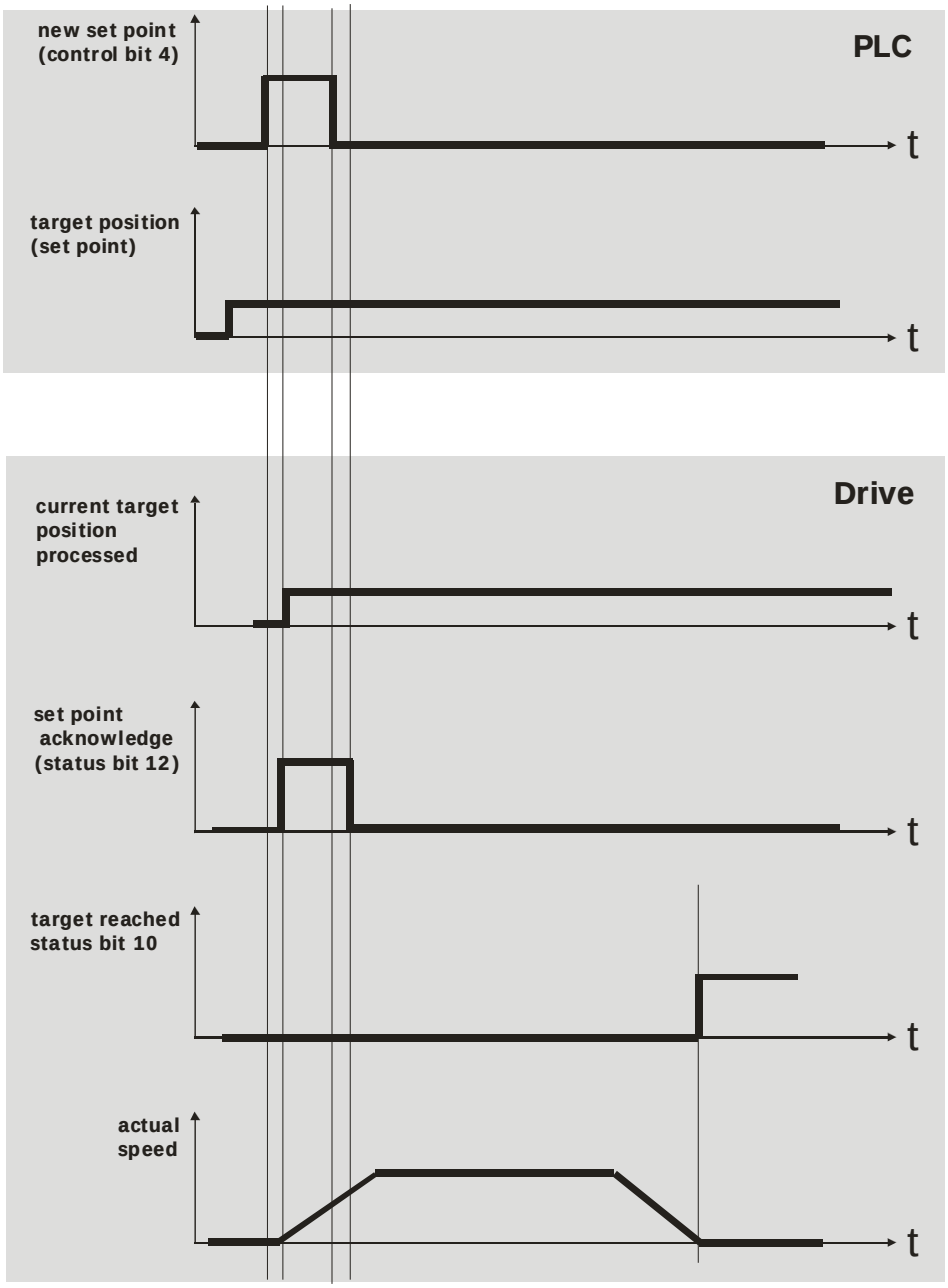
Name	Value	Description
Abs/rel Bit 6	0	*Target position* is an absolute value
	1	*Target position* is a relative value
Halt Bit 8	0	Execute positioning
	1	Stop axle with *profile deceleration* (if not supported with *profile acceleration*), the inverter remains in state “operation enabled”

Status word

Name	Value	Description
Target reached Bit 10	0	Halt (Control word bit 8) = 0: *target position* not reached
		Halt (Control word bit 8)= 1: axle decelerates
	1	Halt (Control word bit 8)= 0: *target position* reached
		Halt (Control word bit 8)= 1: velocity of axle is 0
Set-point acknowledge Bit 12	0	Trajectory generator has not assumed the positioning value (yet)
	1	Trajectory generator has assumed the positioning value
Following error Bit 13	0	No following error
	1	Following error

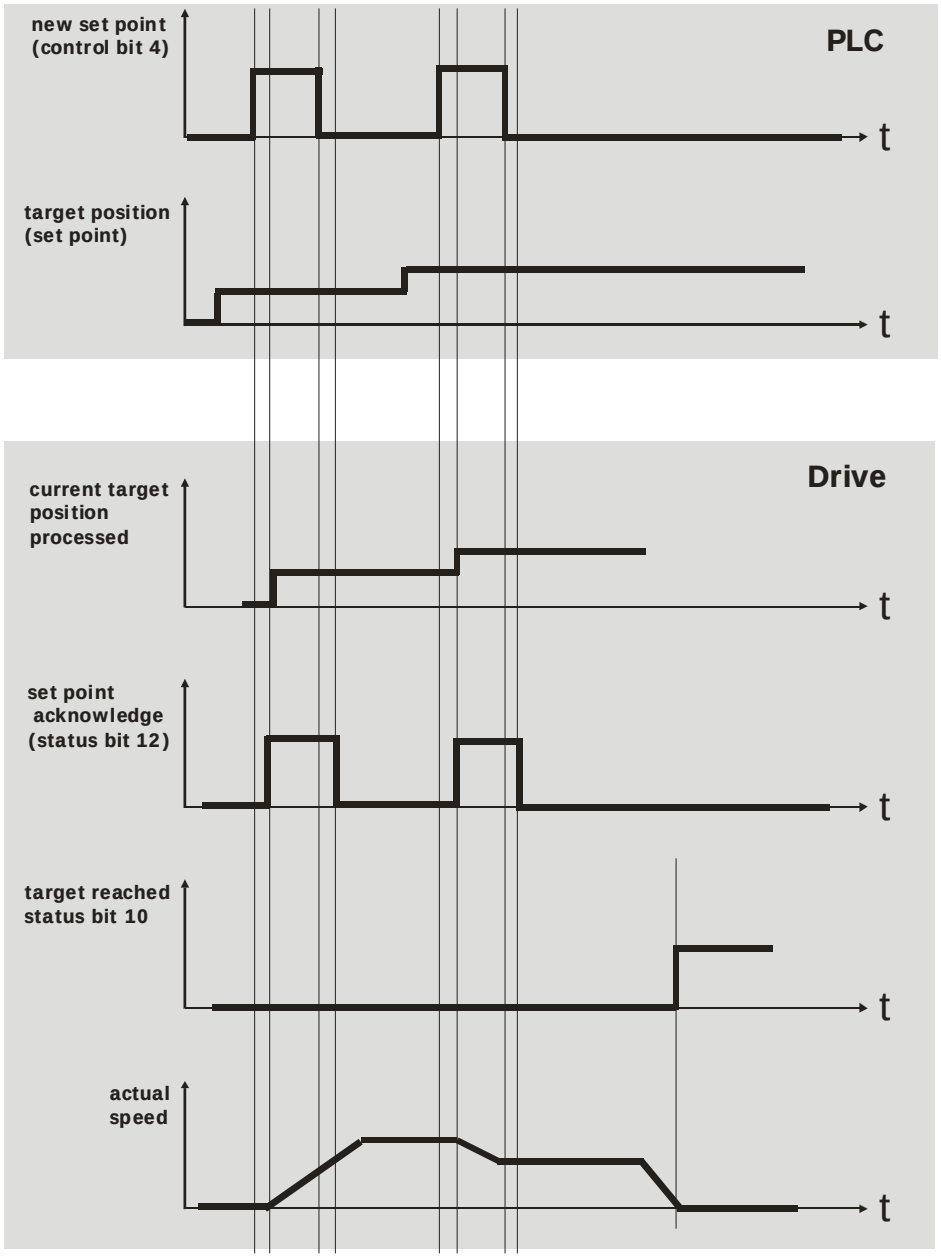
Example: **single set-point**
control bit *change on set-point* = **0**
control bit *change set immediately* = **0**

After a set-point is sent to the drive, the control device signals set-point valid by a rising edge on bit *new set-point* in the Control word. The drive answers by setting bit *set-point acknowledge* and starts moving to the new target position. After that, the control device clears the bit *new set-point* and the drive also clears the bit *set-point acknowledge*. After clearing the bit *set-point acknowledge* the drive is able to accept a new target position.



Example: **single set-point**
control bit *change on set-point* = **0**
control bit *change set immediately* = **1**

When a set-point is in progress and a new set-point is validated by control bit *new set-point* (rising edge), the new set-point is processed immediately.



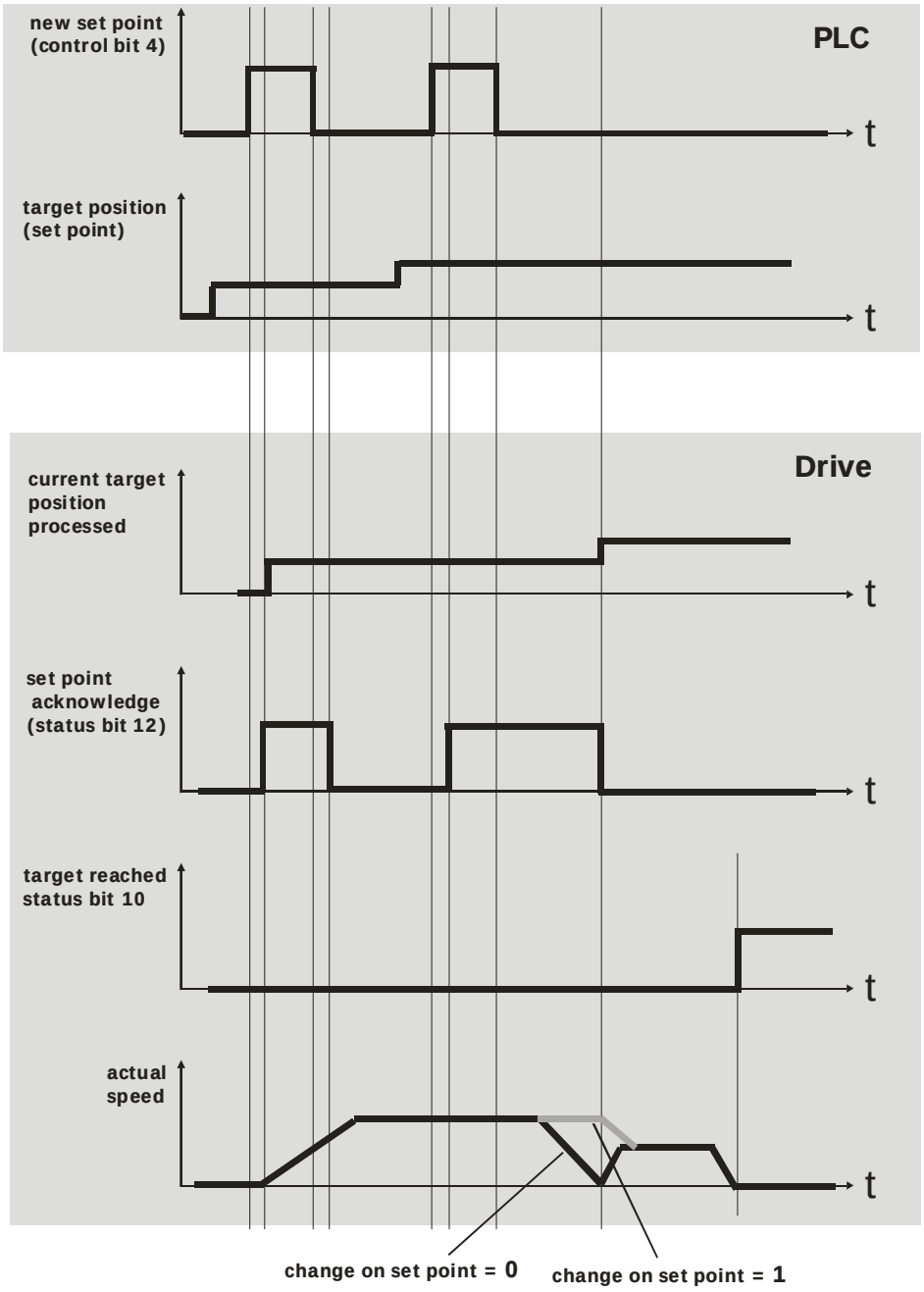
Example: **set of set-points**
control bit *change on set-point* = **0 / 1**
control bit *change set immediately* = **0**

While a Positioning is in progress, the set point is changed.

Change on set point = 0 Current target position will be **stopped** at. After the position is reached, the new set point is taken over.

Change on set point = 1 The current target position will be **driven to in current speed**. As soon as the position is reached, the new set point is taken over.

The grey line in segment "actual speed" shows the actual speed behaviour if control bit *change of set- point* is set (= **1**).



14.4.3.1 Example Sequence

To start the Profile position mode, the correct sequence has to be sent from the PLC.

1	Control word = Status word =	0x0000 0x0050	Disable voltage Switch On Disabled
2	Modes of Operation =	1	(Profile Position mode)
3	Control word = Status word =	0x0006 0x0031	Shutdown Ready to switch on
4	Control word = Status word =	0x0007 0x0033	Switch On Switched On
5	Control word = Status word =	0x0007 ↓ 0x000F 0xnn37	Enable Operation. A positioning is not started Operation enabled
6a	Control word = Status word =	0x0007 or 0x000F → 0x001F 0xnn37	Enable Operation, start with **absolute** movement Profile[1)]. If a motion is already running, that motion is finished, then the new Profile will be used. Operation enabled
6b	Control word = Status word =	0x0007 or 0x000F → 0x005F 0xnn37	Enable Operation, start with **relative** movement Profile[1)]. If a motion is already running, that motion is finished, then the new Profile will be used. Operation enabled
6c	Control word = Status word =	0x0007 or 0x000F → 0x003F 0xnn37	Enable Operation, start with **absolute** movement Profile[1)]. A running motion is changed to the new profile. Operation enabled
6d	Control word = Status word =	0x0007 or 0x000F → 0x007F 0xnn37	Enable Operation, start with **relative** movement Profile[1)]. A running motion is changed to the new profile. Operation enabled
7	Control word = Status word =	0x01nF 0xnn37	HALT: The drive is stopped with ramp 0x6049 *Velocity deceleration*. Operation enabled

1) A profile consists of the following entries. If a value is not changed, the old value will still be active.

- 0x607A *Target Position*
- 0x6081 *Profile velocity*
- 0x6083 *Profile acceleration*
- 0x6084 *Profile deceleration*

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

To start a new Position Profile, it is not necessary to change the Control word to 0x0007 first and switch to 0xnnnF.
After a position profile is finished a new Profile can be started from Control word 0xnnnF by using the “New Setpoint” Bit (Bit 4).
While a position profile is active, using the “Change Setpoint immediately” (Bit 5) and “New Setpoint” (Bit 4) will start a new profile without stopping.

14.4.4 Interpolated position mode

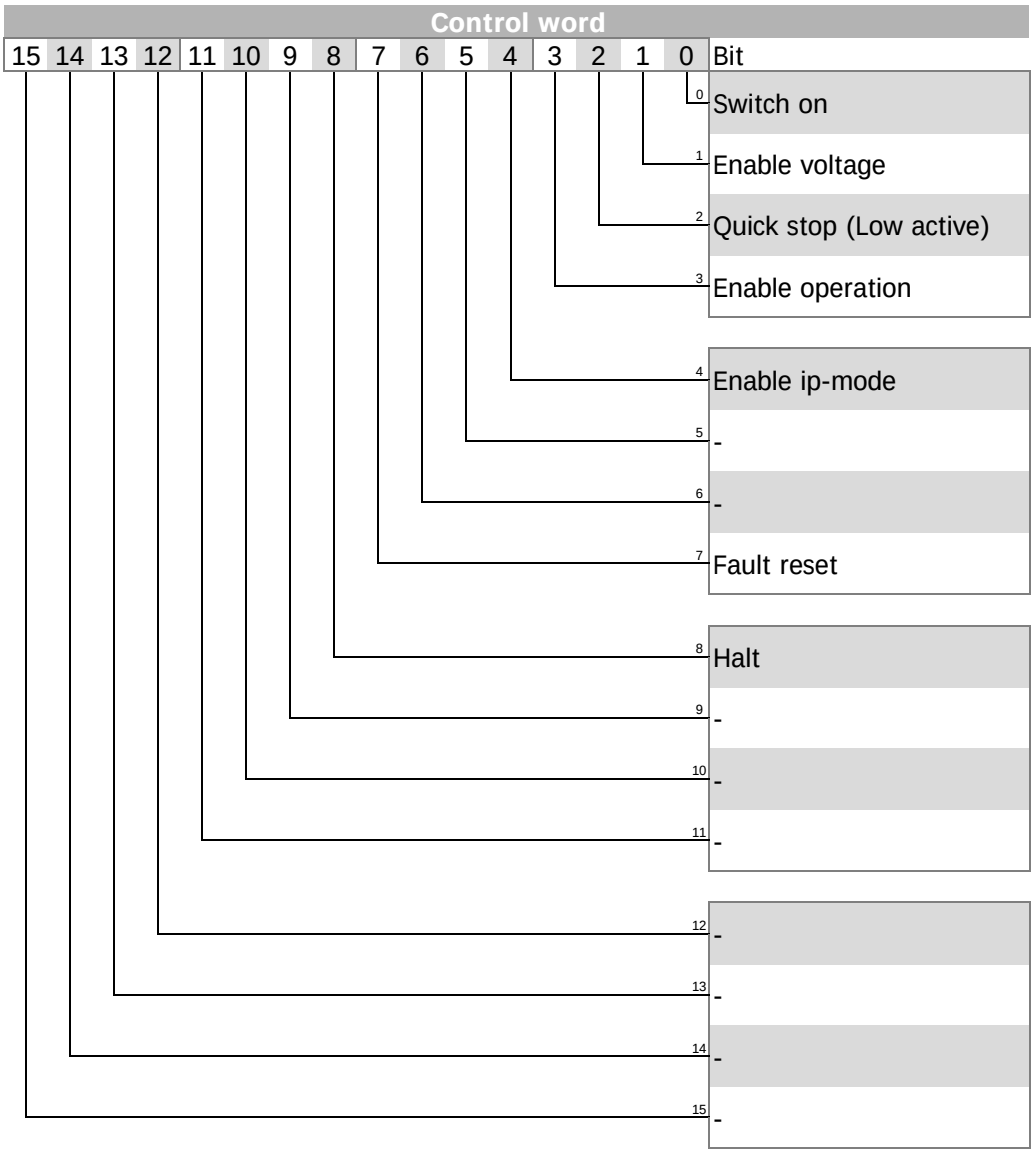
The interpolated position mode is selected via object 0x6060/0 *Modes of operation* = **7**. In interpolated position mode the inverter receives a target position at equidistant time intervals.

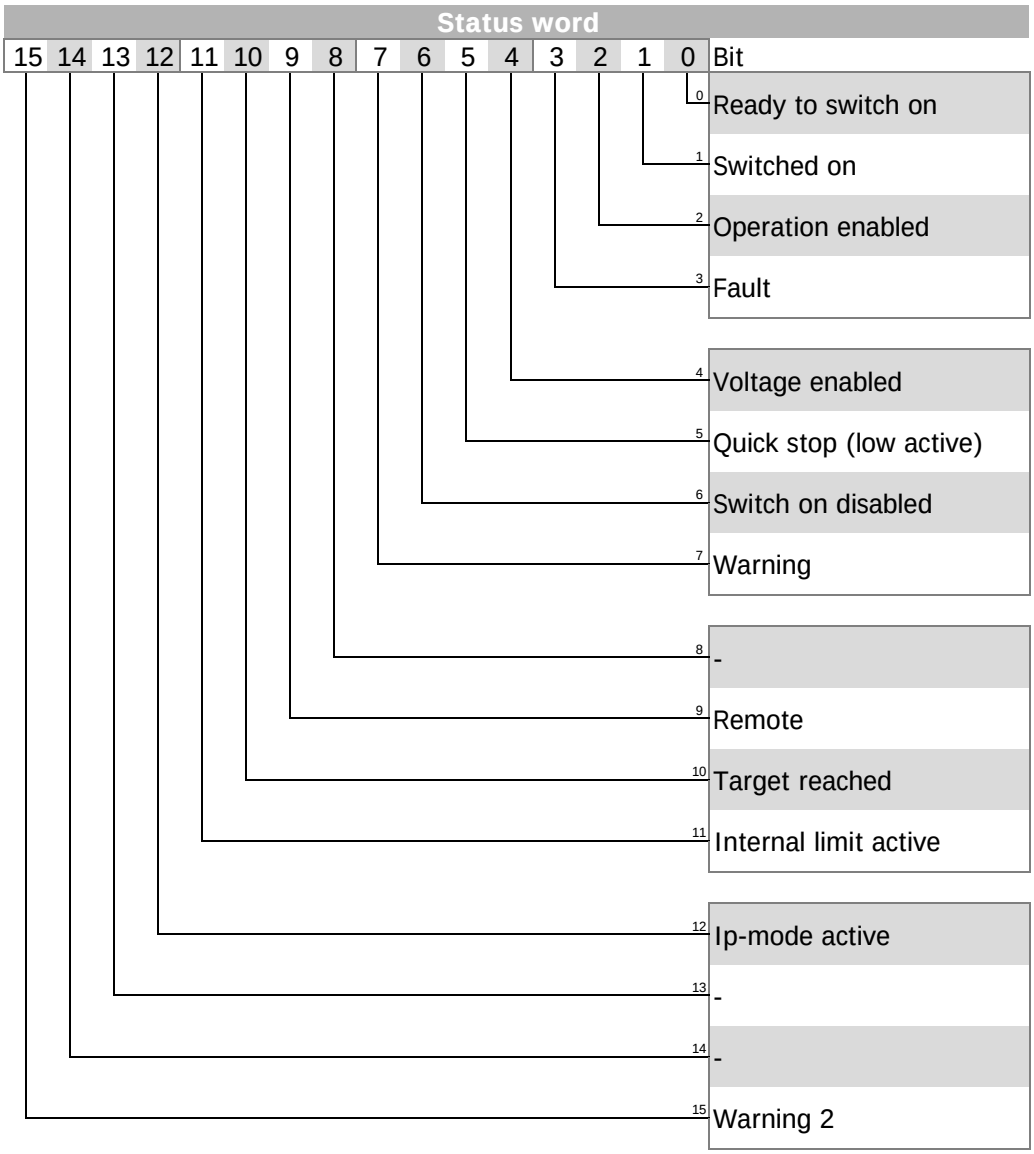
Related objects:

0x6040	Control word
0x6041	Status word
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x6083	Profile acceleration
0x6084	Profile deceleration
0x6085	Quick stop deceleration
0x60C1	Interpolation data record

The Ramp Rise/Fall times are set up via parameters **1176** and **1178** and object 0x6086.

In interpolated position mode the "operation mode specific" bits of *Control word* and *Status word* are used as shown:





Only linear interpolation is available (for this, object 0x60C0/0 *interpolation submode select* is NOT implemented). For proper operation, object 0x60C1/1 *interpolation data re*cord must be sent by a synchronous RxPDO. The time interval of the SYNC messages MUST be constant. For the evaluation of the SYNC interval, object 0x1006/0 *communication cycle period* must be set or at least eleven SYNC messages are necessary before the interpolated position mode is activated by setting control bit 4 *enable_ip_mode*. For the SYNC interval only multiples of milliseconds are allowed (1, 2, 3, 4, ...; refer to chapter 12.2.4 “0x1006/0 Communication Cycle Period”).
The activation of interpolated position mode is displayed by status bit 12 *ip_mode_active*.
With each SYNC message a new target position is transmitted to the drive by object 0x60C1/1 *interpolation data record*. The new interpolated reference positions and an additional speed reference are calculated from the last reference position, the *interpolation data record* and the time interval of the SYNC messages. As shown, the target position actually received will be reached at the time of the next SYNC message.

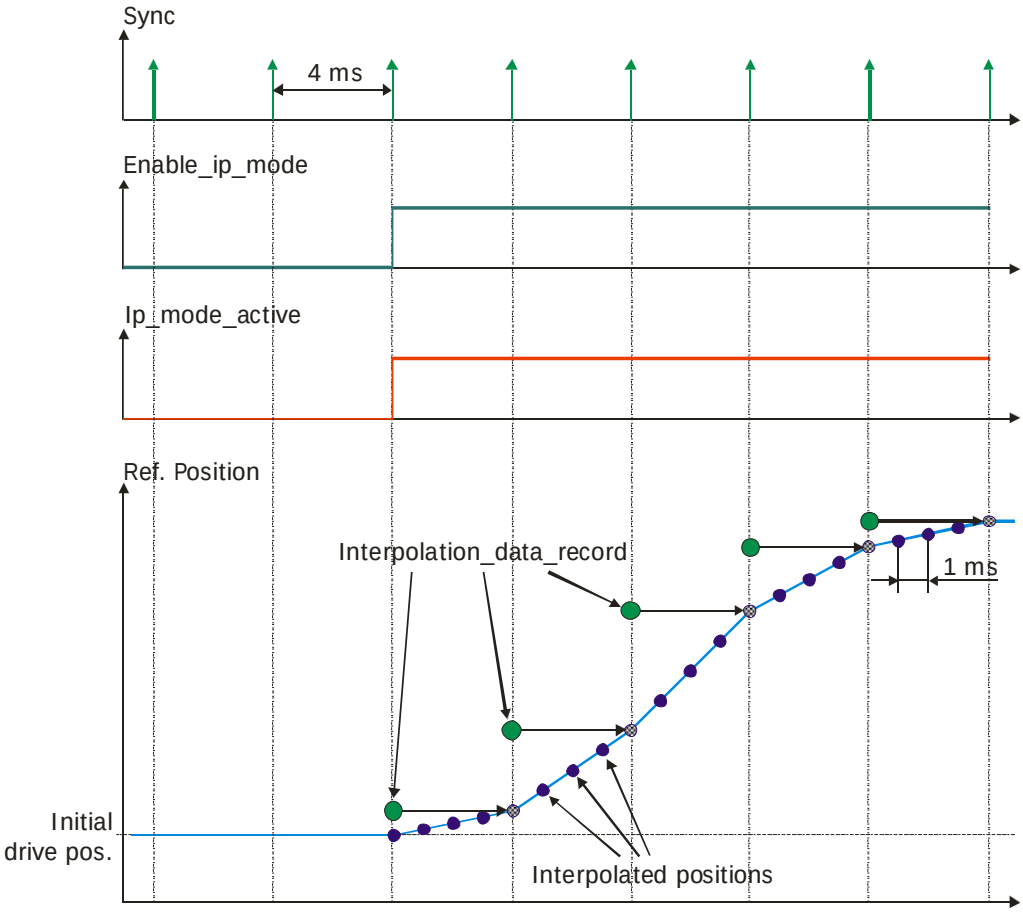
- 0x6083 *Profile acceleration* is only used when the interpolated mode is activated (rising edge of Bit 4 “enable ip-mode”).Then this acceleration is used to synchronize from the actual speed to the calculated speed of the interpolated trajectory.
- 0x6084 *Profile deceleration* is used when the interpolated mode is switched off (falling edge of Bit 4 “enable ip-mode”) or a HALT signal (Bit 8) is set.
- 0x6085 *Quick stop deceleration* or 0x6084 *Profile deceleration* is used when a fault occurred. This can be changed via Stopping behaviour **630** *Operation mode* and Communication fault reaction 0x6007/0 *abort connection option code*.
- Please set the acceleration values big enough, so that for switching on and switching off of the IP mode the motion is synchronized with the PLC.

Control word

Name	Value	Description
Enable IP-mode **Bit 4**	0	Interpolated position mode inactive
	1	Interpolated position mode active
Halt **Bit 8**	0	Execute the instruction of bit 4 “enable ip-mode”
	1	Stop axle , inverter remains in state “operation enabled” 0x6084 *Profile deceleration* is used for deceleration.

Status word

Name	Value	Description
Target reached **Bit 10**	0	Halt (Control word bit 8)= 0: position not (yet) reached
		Halt (Control word bit 8)= 1: axle decelerates
	1	Halt (Control word bit 8)= 0: position reached
		Halt (Control word bit 8)= 1: axle has velocity 0
IP-mode active **Bit 12**	0	Interpolated position mode inactive
	1	Interpolated position mode active


Sync
4 ms
Enable_ip_mode
Ip_mode_active
Ref. Position
Interpolation_data_record
1 ms
Initial
drive pos.
Interpolated positions

14.4.4.1 Example Sequence

To start the Interpolated position mode, the correct sequence has to be sent from the PLC.

1	Control word = Status word =	0x0000	0x0050	Disable voltage Switch On Disabled
2	Modes of Operation =	7		(Interpolated Positon mode)
3	Control word = Status word =	0x0006	0x0031	Shutdown Ready to switch on
4	Control word = Status word =	0x0007	0x0033	Switch On Switched On
5a	Control word = Status word =	0x000F	0xnn37	Enable Operation. Operation enabled
5b	Control word = Status word =	0x001F	0x1n37	Enable Operation and start Interpolated Mode (IP). Operation enabled

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

Always ensure that a valid Position is contained in “Interpolated Data Record”. It is recommended to copy the Actual Position into the “Data Record” before starting the Interpolated Mode.

14.4.5 Homing mode

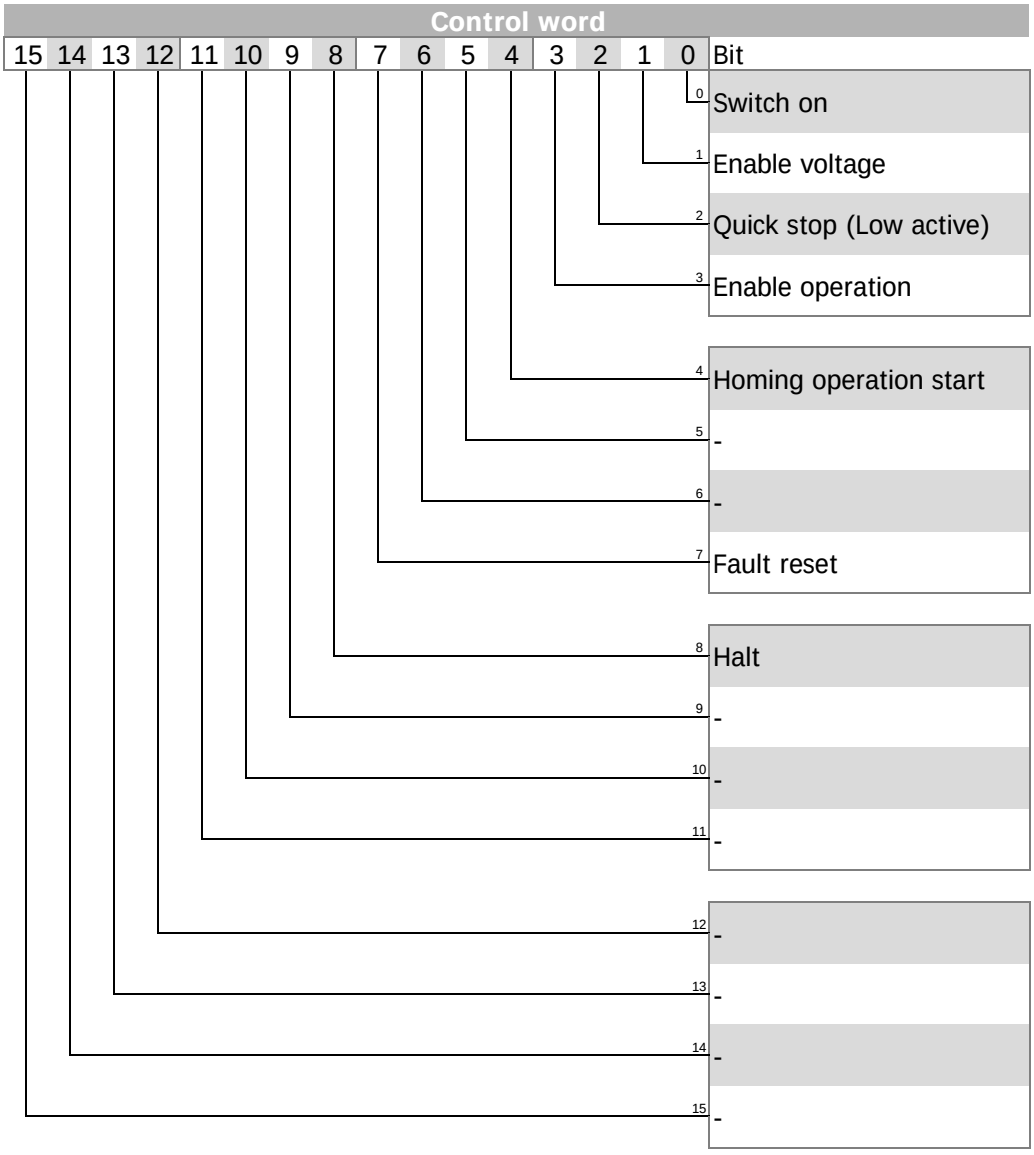
The homing mode is selected via object 0x6060/0 *Modes of operation* = **6**. In homing mode the inverter moves the drive to a reference position. The method used for this movement is defined by object 0x6098 *homing method*.

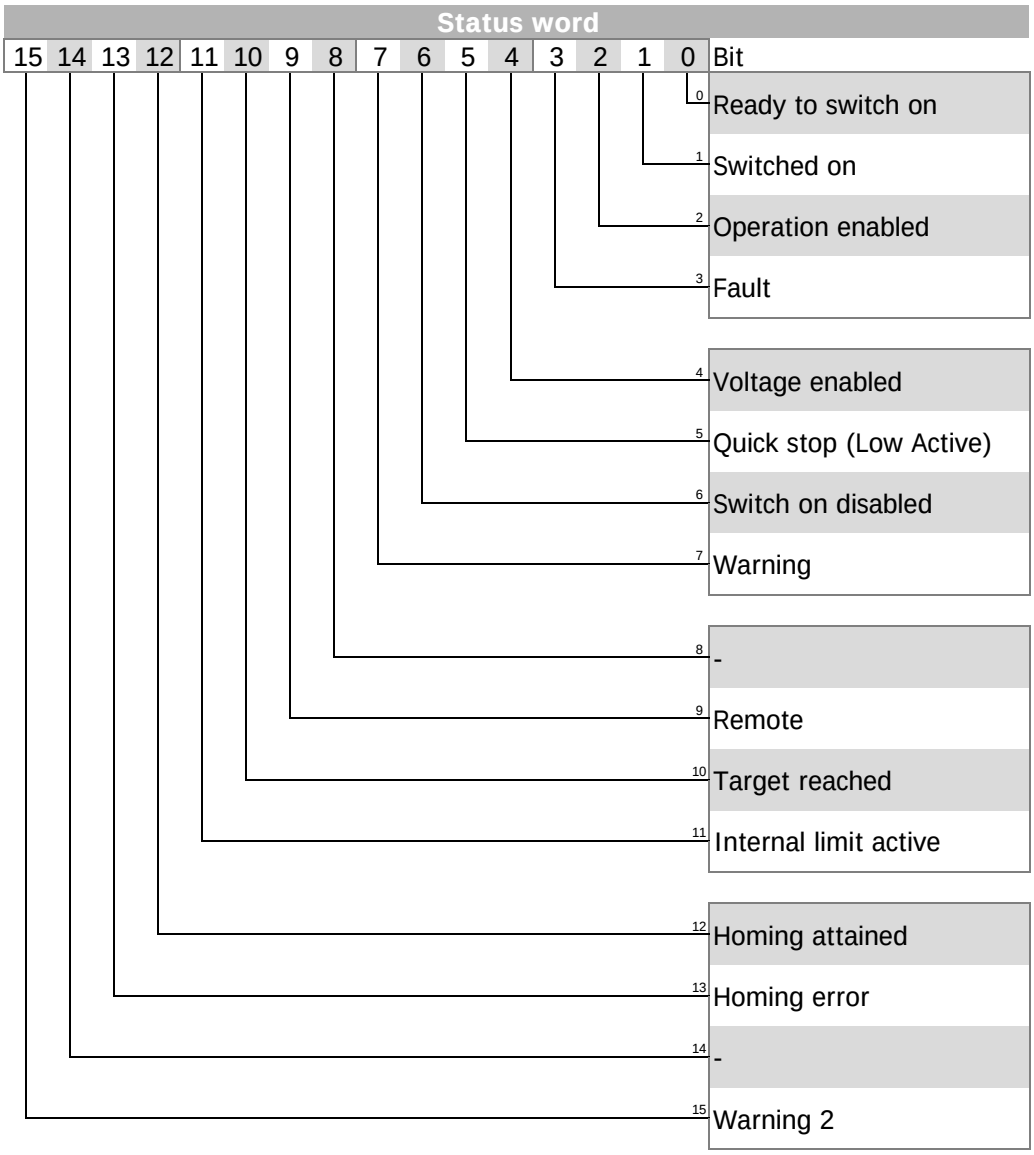
Related objects:

0x6040	Control word
0x6041	Status word
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x6098	Homing method
0x6099	Homing speeds
0x609A	Homing acceleration

The Ramp Rise/Fall times are set up via parameters **1135**.

In homing mode the "operation mode specific" bits of *Control word* and *Status word* are used as shown:





Control word

Name	Value	Description
Homing operation start **Bit 4**	0	Homing mode inactive
	0 → 1	Start homing mode
	1	Homing mode active
	1 → 0	Interrupt homing mode
Halt **Bit 8**	0	Execute instruction of bit 4 “homing operation start”
	1	Stop axle with homing acceleration, , inverter remains in state “operation enabled”

Status word

Name	Value	Description
Target reached **Bit 10**	0	Halt = 0: home position not reached
		Halt = 1: axle decelerates
	1	Halt = 0: home position reached
		Halt = 1: axle has velocity 0
Homing attained **Bit 12**	0	Homing not yet completed
	1	Homing mode carried out successfully
Homing error **Bit 13**	0	No homing error
	1	Homing error occurred Homing mode carried out unsuccessfully

For an exact description of the various homing modes refer to the application manual "Positioning".

14.4.5.1 Example Sequence

To start the Homing mode, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000	Disable voltage
	Status word =	0x0050	Switch On Disabled
2	Modes of Operation =	6	(Homing)
3	Control word =	0x0006	Shutdown
	Status word =	0x0031	Ready to switch on
4	Control word =	0x0007	Switch On
	Status word =	0x0033	Switched On
5a	Control word =	0x000F	Enable Operation.
	Status word =	0xnn37	Operation enabled
5b	Control word =	0x001F	Enable Operation and start Homing.
	Status word =	0x1n37	Operation enabled and homing attained.

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In "Operation enabled" state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

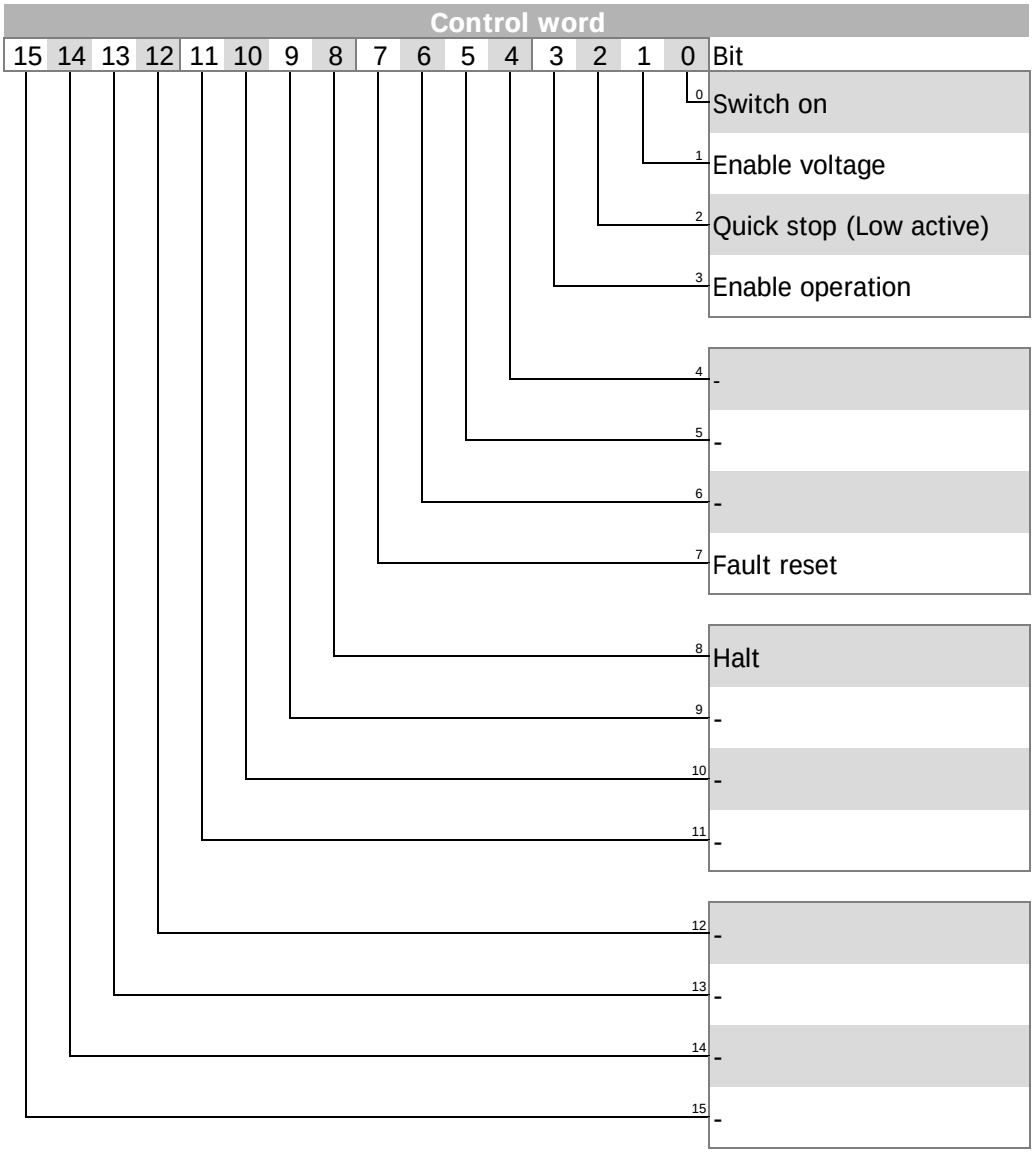
14.4.6 Cyclic Synchronous position mode

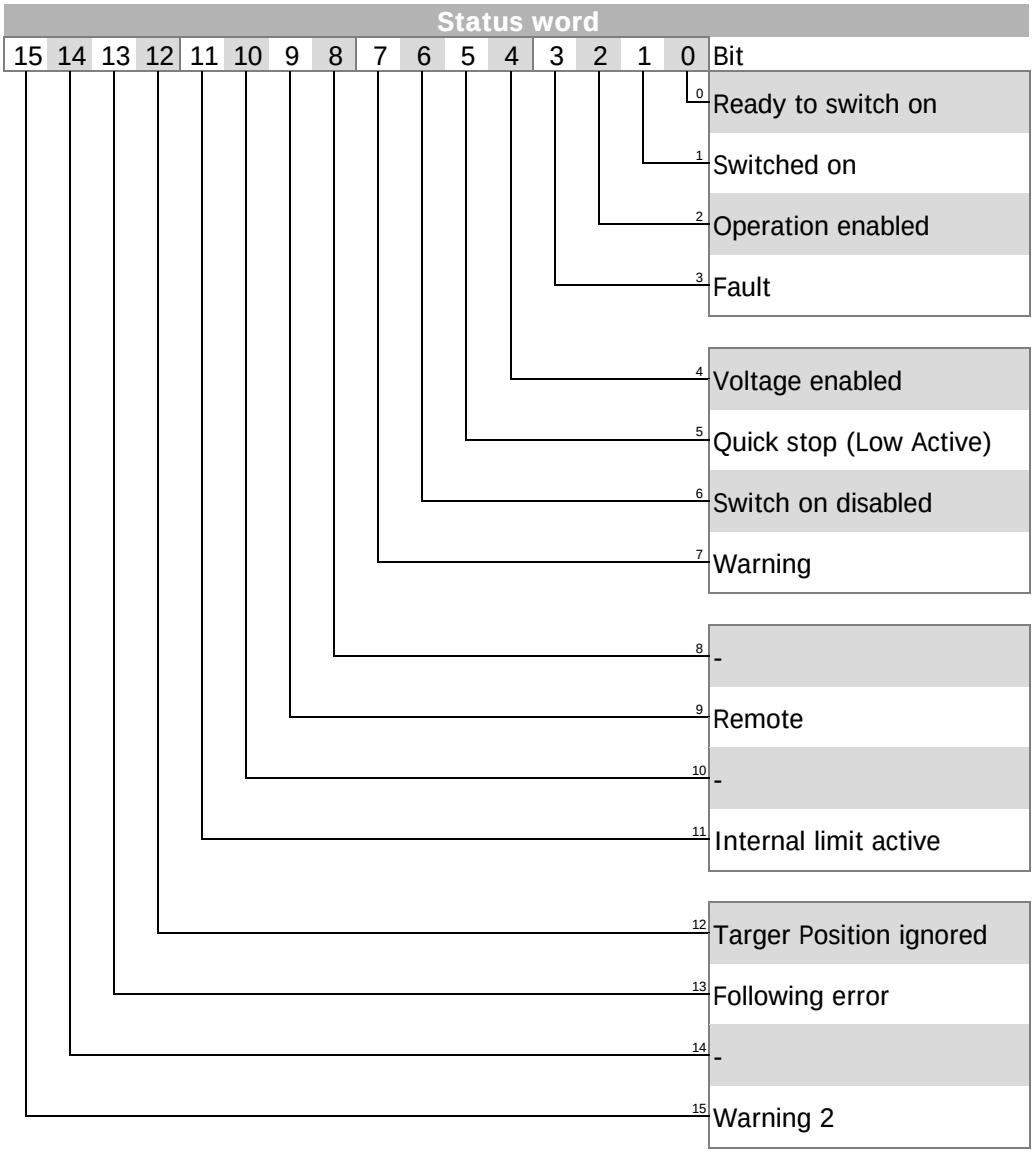
The Cyclic Synchronous position mode is selected via object 0x6060/0 *Modes of operation* = **8**. In Cyclic Synchronous position mode the inverter receives a target position at equidistant time intervals.

Related objects:

0x6040	Controlword
0x6041	Statusword
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x607A	Target Position
0x6085	Quick stop deceleration

In Operation mode *Cyclic Synchronous position mode* only the 4 lowest bity are used for control.





Status word

Name	Value	Description
Target position ignored	0	Target Position is ignored.
Bit 12	1	Target Position is used as Reference value.
Following error	0	No following error
Bit 13	1	Following error

No ramp limits are active inside the frequency inverter. Limit the dynamic actions suitable by the PLC.

- First copy before the Start inside the SPS the actual Position 0x6064 to the target position.
- Start the Control Sequency in the PLC (0x0,0x6,0x7, 0xF).
- Wait until in the Status word Bit 12 is active.
- Now update the target Position in the PLC program.

14.4.6.1 Example Sequence

To start “Cyclic synchronous position mode”, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000	Disable voltage
1	Status word =	0x0050	Switch On Disabled
2	Modes of Operation =	8	(Cyclic synchronous position mode)
3	Control word =	0x0006	Shutdown
	Status word =	0x0031	Ready to switch on
4	Control word =	0x0007	Switch On
	Status word =	0x0033	Switched On
5	Control word =	0x000F	Enable Operation.
	Status word =	0xnn37	Operation enabled

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
With the control word transition from 0xnnnF to 0x0007 the “Cyclic Synrchonous Position mode” is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

Always ensure that a valid Position is contained in “Target Position”. It is recommended to copy the Actual Position into the “Target Position” before starting.

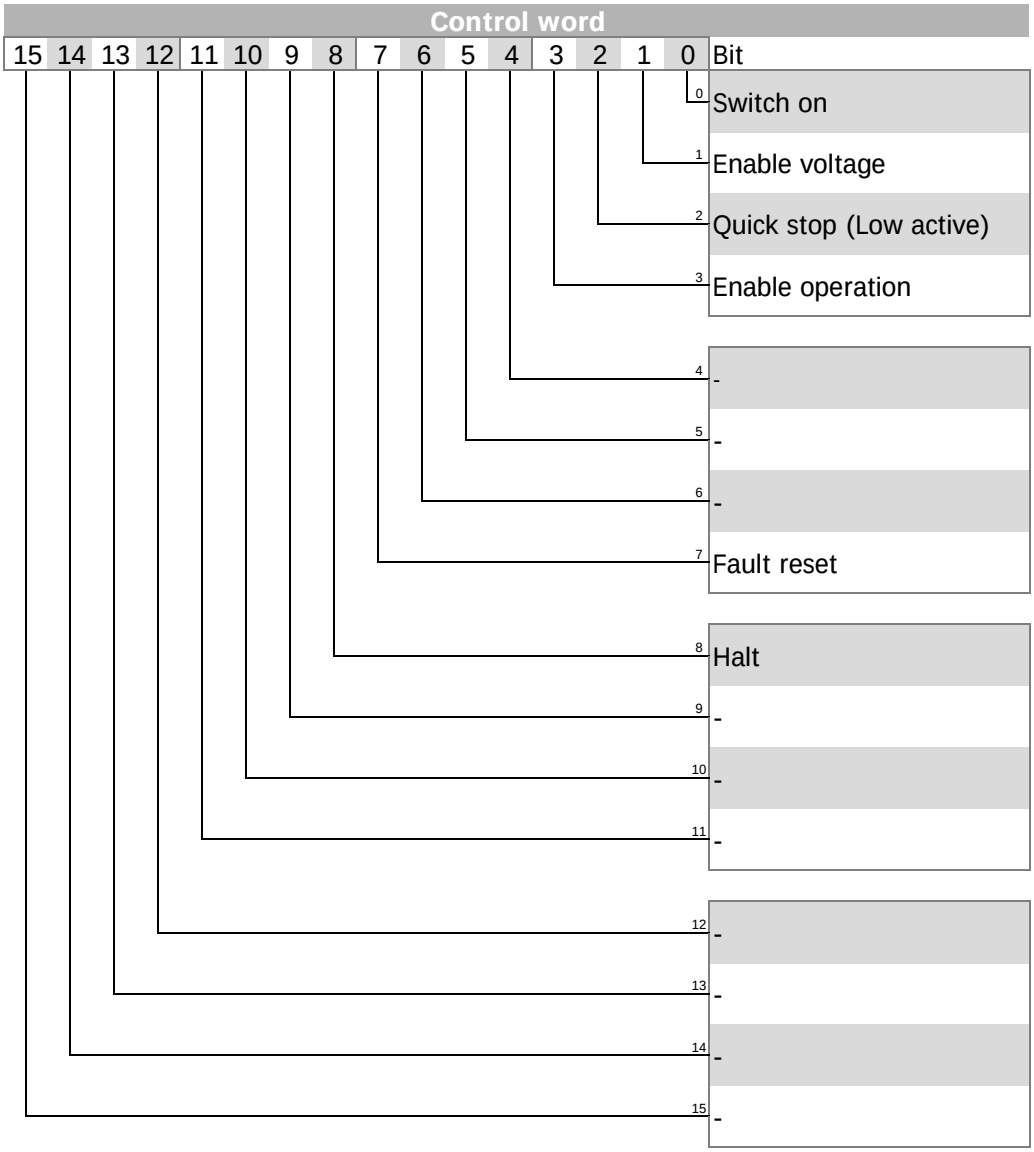
14.4.7 Cyclic Synchronous Velocity mode

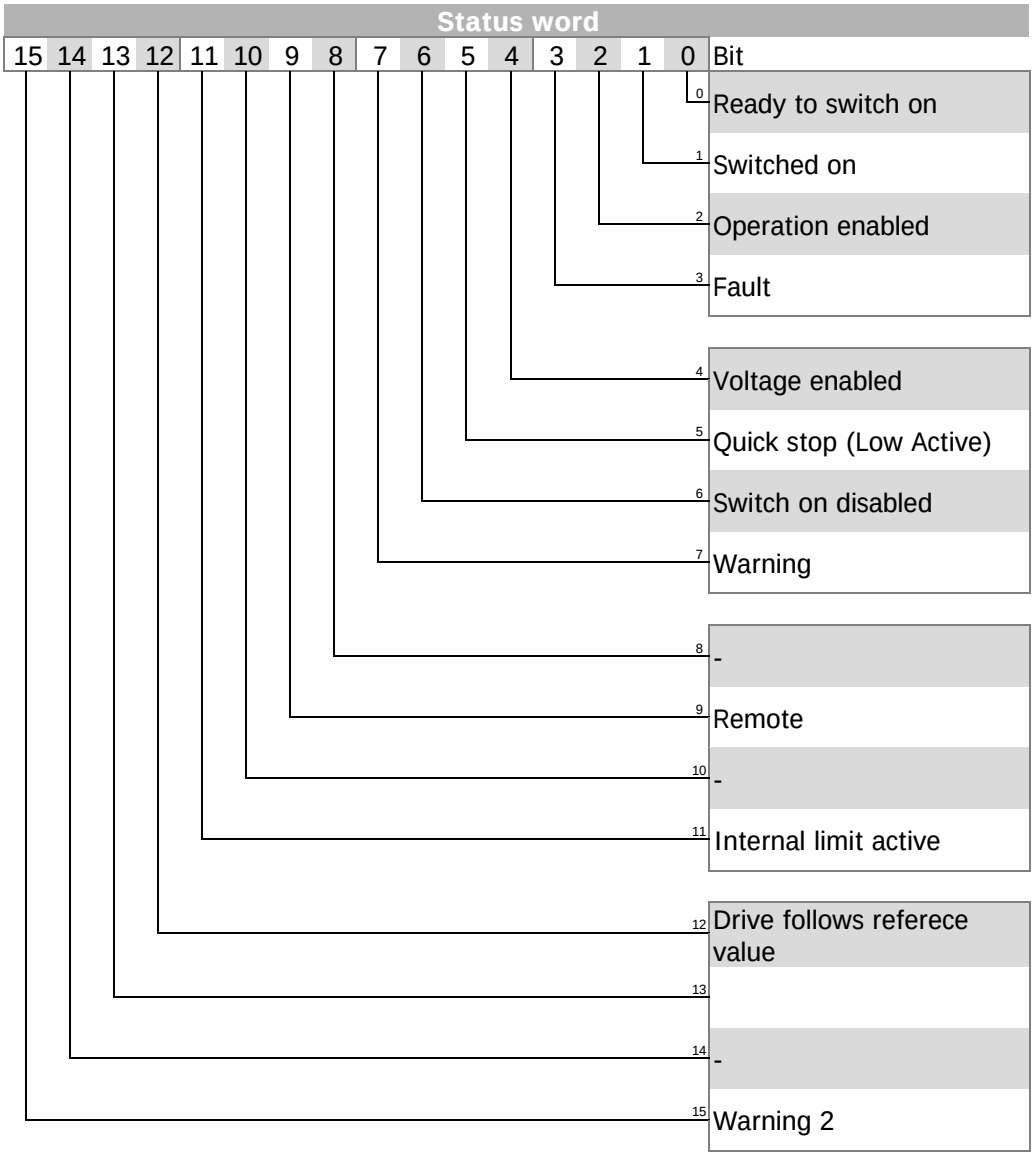
The Cyclic Synchronous Velocity mode is selected via object 0x6060/0 *Modes of operation* = **9**. In Cyclic Synchronous Velocity mode the inverter receives a reference speed in equidistant time intervals.

Related objects:

0x6040	Controlword
0x6041	Statusword
0x6046	Velocity min max amount
0x6060	Modes of operation
0x6061	Modes of operation display
0x6085	Quick stop deceleration
0x60FF	Target Velocity

In Operation mode *Cyclic Synchronous position mode* only the 4 lowest bity are used for control.





Status word

Name	Value	Description
Drive follows referece value	0	Drive does not follow the reference value.
Bit 12	1	Drive follows the reference value.
Following error	0	No following error
Bit 13	1	Following error

No ramp limits are active inside the frequency inverter. Limit the dynamic actions suitable by the PLC.

- Start the Control Sequency in the PLC (0x0,0x6,0x7, 0xF).
- Wait until in the Status word Bit 12 is active.
- Now update the Reference speed in the PLC program.

14.4.7.1 Example Sequence

To start “Cyclic Synchronous Velocity mode”, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000		Spannung sperren
1	Status word =		0x0050	Einschalten gesperrt
2	Modes of Operation =	9		(Cyclic Synchronous Velocity mode)
3	Control word =	0x0006		Stillsetzen
	Status word =		0x0031	Einschaltbereit
4	Control word =	0x0007		Einschalten
	Status word =		0x0033	Eingeschaltet
5	Control word =	0x000F		Betrieb freigeben.
	Status word =		0xnn37	Betrieb freigegeben

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
With the control word transition from 0xnnnF to 0x0007 the “Cyclic Synrchonous Position mode” is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

Always ensure that a valid Position is contained in “Target Position”. It is recommended to copy the Actual Position into the “Target Position” before starting.

14.4.8 Table travel record

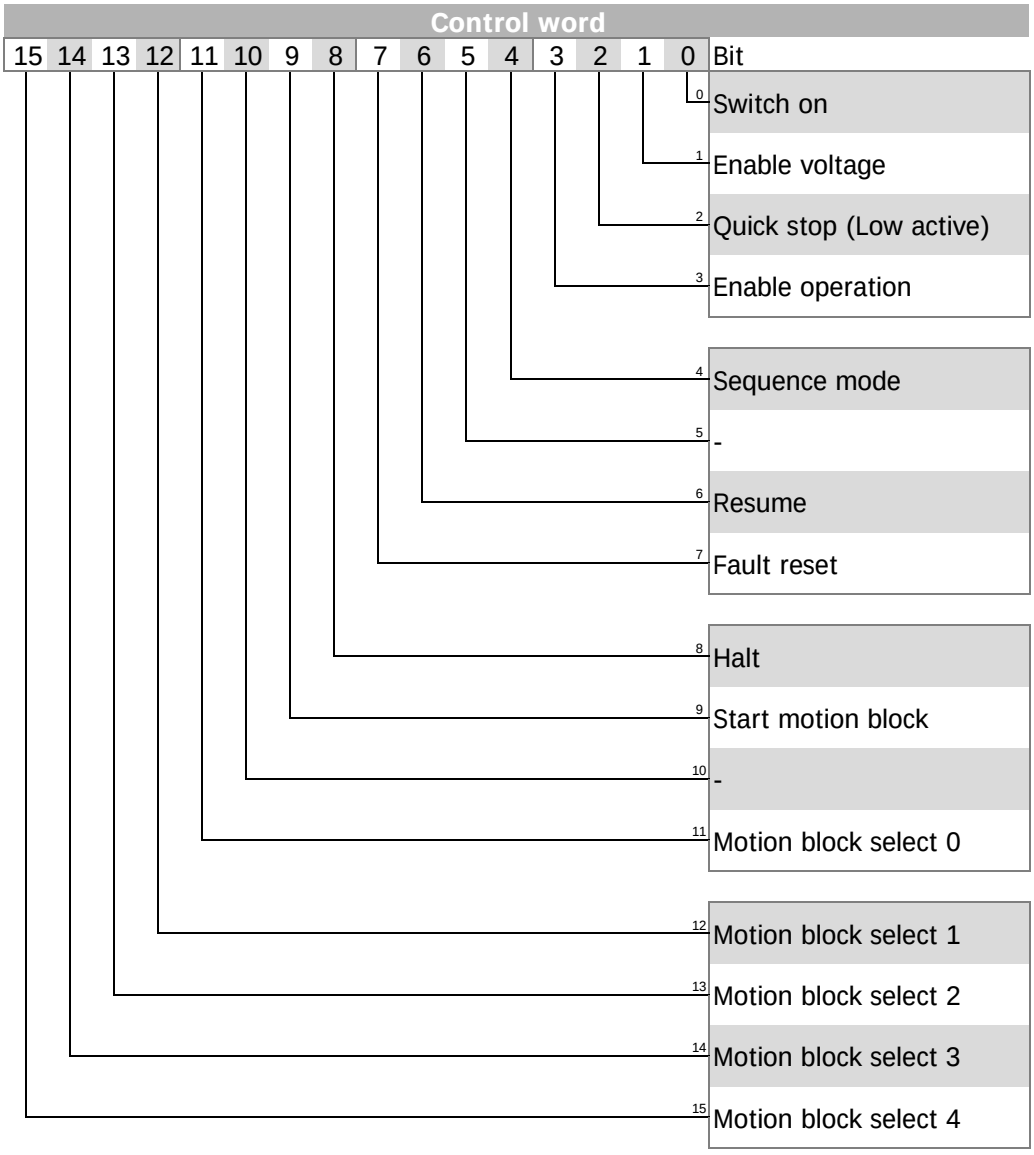
The table travel record mode is selected via object 0x6060/0 *Modes of operation* = **0xFF** =**-1**. In table travel record mode the inverter moves the drive autonomous to consecutive positions.
The table travel record mode uses predefined sets of positions. Every target position is defined by one motion block. Several sets of motion blocks may be defined.

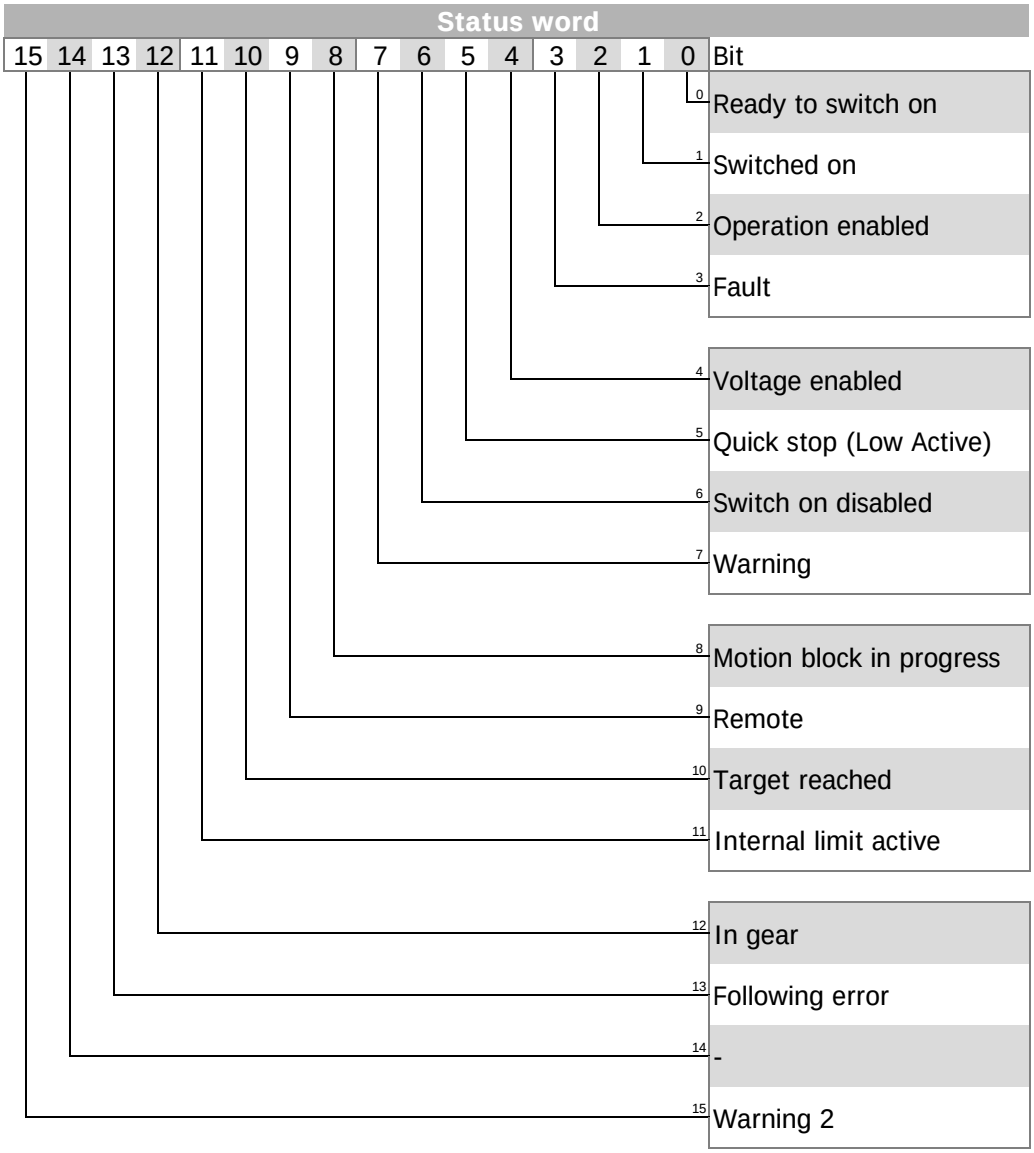
For an exact description of table travel record mode options refer to the application manual “Positioning”.

Related objects:

0x6040	Control word	0x6064	Position actual value
0x6041	Status word	0x6065	Following error window
0x6046	Velocity min max amount	0x6066	Following error time
0x6060	Modes of operation	0x6067	Position window
0x6061	Modes of operation display	0x6068	Position window time
0x5FF0	Active motion block	0x6085	Quick stop deceleration
0x5FF1	Motion block to resume		

In table travel mode the "operation mode specific" and “manufacturer specific” bits of *Control word* and *Status word* are used as shown:




Status word
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit
0 Ready to switch on
1 Switched on
2 Operation enabled
3 Fault
4 Voltage enabled
5 Quick stop (Low Active)
6 Switch on disabled
7 Warning
8 Motion block in progress
9 Remote
10 Target reached
11 Internal limit active
12 In gear
13 Following error
14 -
15 Warning 2

Control word

Name	Value	Description
Sequence mode **Bit 4**	0	Single motion block
	1	Sequence of motion blocks
Resume **Bit 6**	0	Start motion block = motion block select
	1	Start motion block = last active motion block
Halt **Bit 8**	0	Execute instruction of bit 4 “sequence mode”
	1	Stop axle with ramp of actual motion block, inverter remains in state “operation enabled”
Start motion block **Bit 9**	0	Stop axle with ramp of actual motion block
	0 → 1	Start execution of motion block(s)
Motion block select 0...4 **Bit 11...15**	n	Start motion block = n + 1

Motion block select:

Control word															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Motion block select						Sta	Halt		Res		Seq				
4	**3**	**2**	**1**	**0**											

Start motion block = motion block select + 1:

Motion block select					resulting start motion block
4	3	2	1	0	
0	0	0	0	0	1
0	0	0	1	1	4
1	0	0	0	0	17
1	1	1	1	1	32

Status word

Name	Value	Description
Motion block in progress **Bit 8**	0	Single motion block: motion block completed Sequence of motion blocks: sequence completed
	1	Single motion block/sequence of motion blocks active
Target reached **Bit 10**	0	Halt = 0: target position not reached yet (motion blocks with positioning only)
		Halt = 1: axle decelerates
	1	Halt = 0: target position reached (motion blocks with positioning only)
		Halt = 1: axle has velocity 0
In gear **Bit 12**	0	Electronic gear not coupled
	1	Electronic gear coupled
Following error **Bit 13**	0	No following error
	1	Following error

Basic functions

The control bit *sequence mode* decides between execution of one single motion block (*sequence mode = 0*) and execution of a sequence of motion blocks (*sequence mode = 1*).

In both cases the selection of the desired motion block (motion block number of single motion block or start motion block number of motion block sequence) is done by calculating the motion block number from *motion block select* with the rising edge of *start motion block*

While the selected motion block or motion block sequence is processed *motion block in progress* is set to 1. *Motion block in progress* remains 1 until the motion block processing is finished. When using a single motion block *motion block in progress* is set to 0 with completion of the single motion block. Otherwise when using a motion block sequence *motion block in progress* is set to 0 when reaching a next motion block setting of 0 (end of motion block) or -1 (error) -2 (stop and error) or -3 (quick stop and error).

While processing a motion block sequence the actual processed motion block is displayed by object 0x5FF0 *active motion block*.

If motion block processing is interrupted by setting *start motion block* to 0 the axle stops with the ramp defined by the actual motion block. The interrupted motion block/motion block sequence can be restarted again by setting *resume* and a rising edge of *start motion block*.

If *resume* is set to 1 and there is no valid motion block available the motion block defined by *motion block select* will be used. A valid motion block is displayed by object 0x5FF1 *motion block to resume*. *Motion block to resume* displays -1 in case there is no valid motion block or the last motion block/motion block sequence was not interrupted.

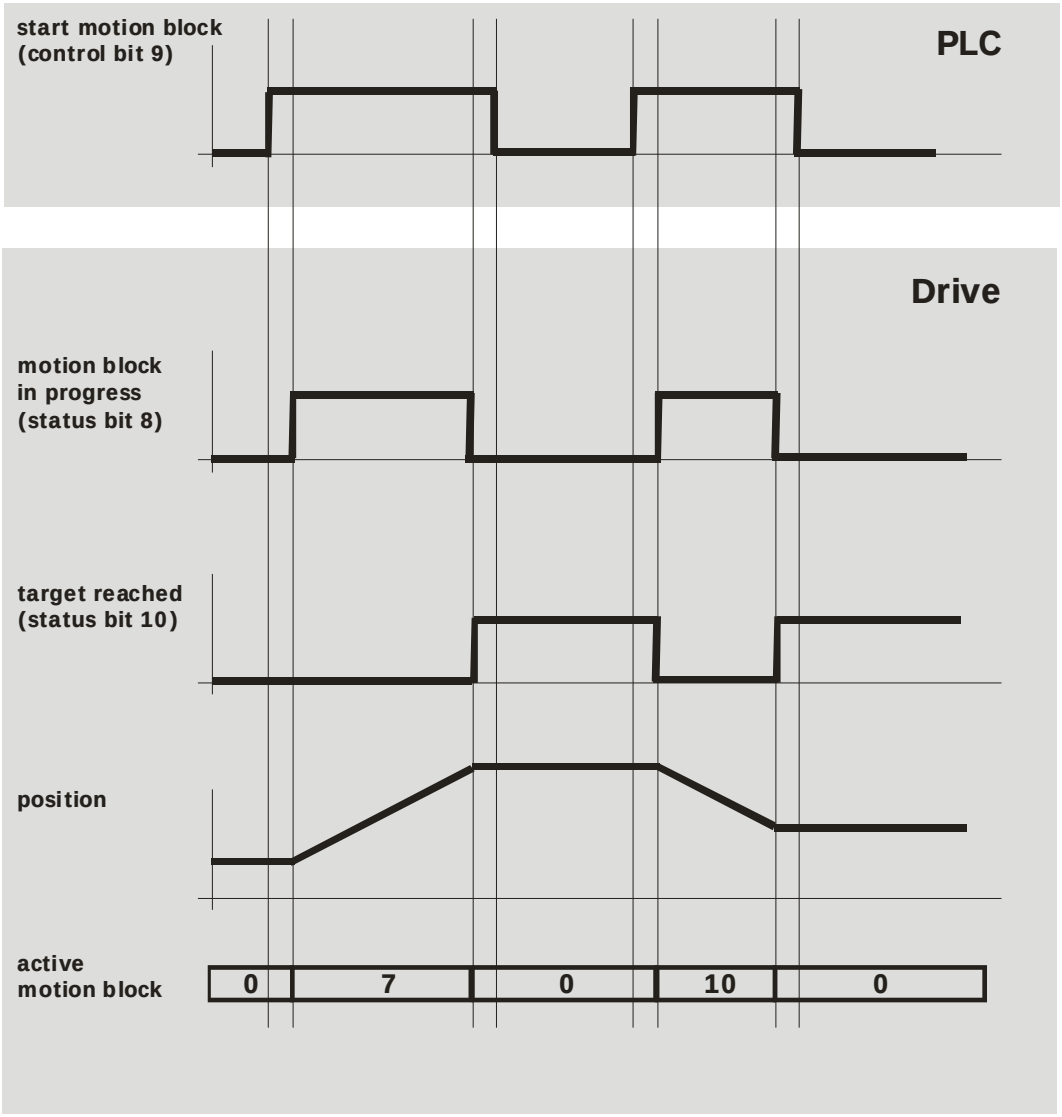
Target reached is set by motion blocks with absolute or relative positioning when the actual position reaches the *position window*.

In gear is set when the function electronic gear is used and the gear is coupled.

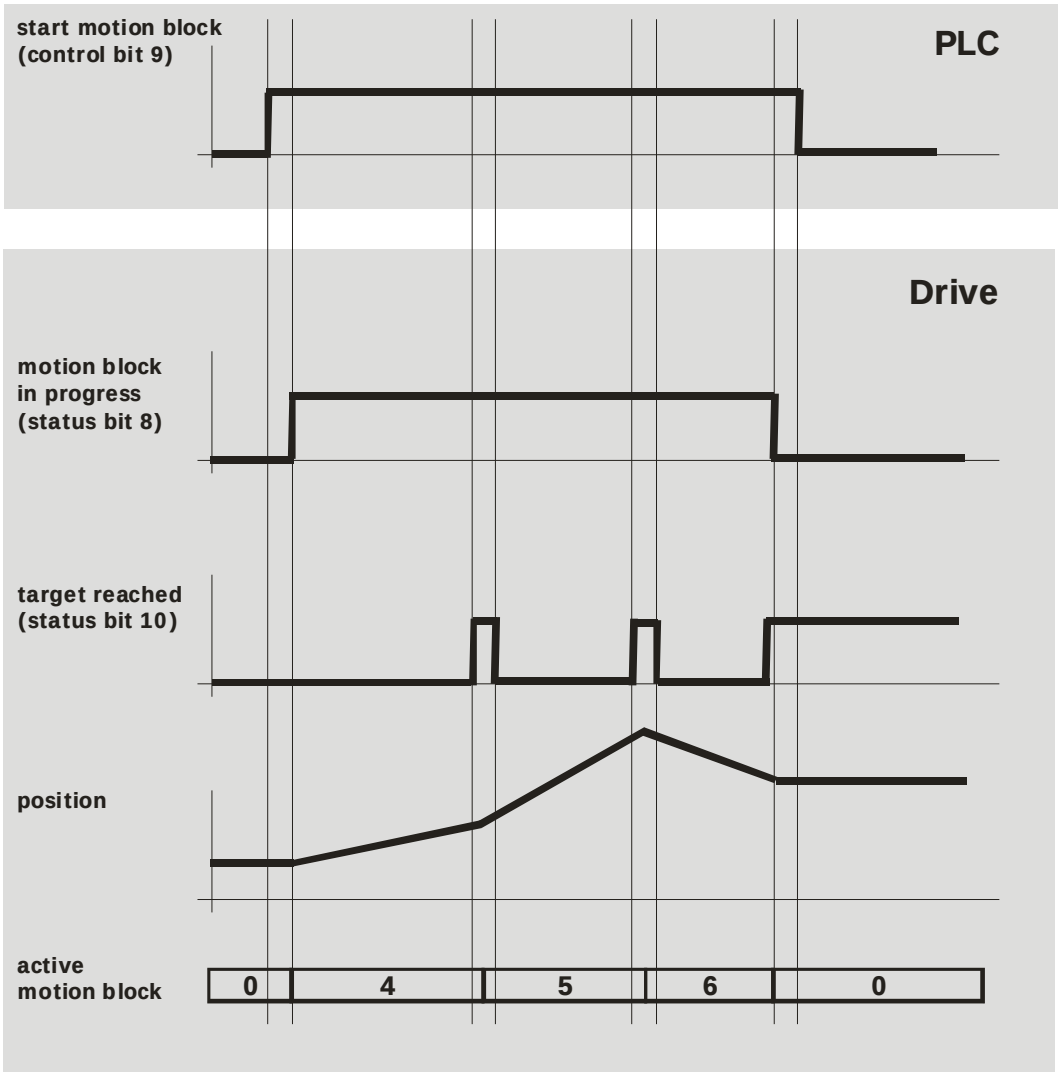
Setting *Halt* to 1 interrupts an actual processed motion block. The axle is stopped with the ramp defined by the actual motion block. When reaching velocity 0 *target reached* is set to 1. The drive remains in state *operation enabled*. Resetting *Halt* to 0 restarts processing of the interrupted motion block.

Examples:

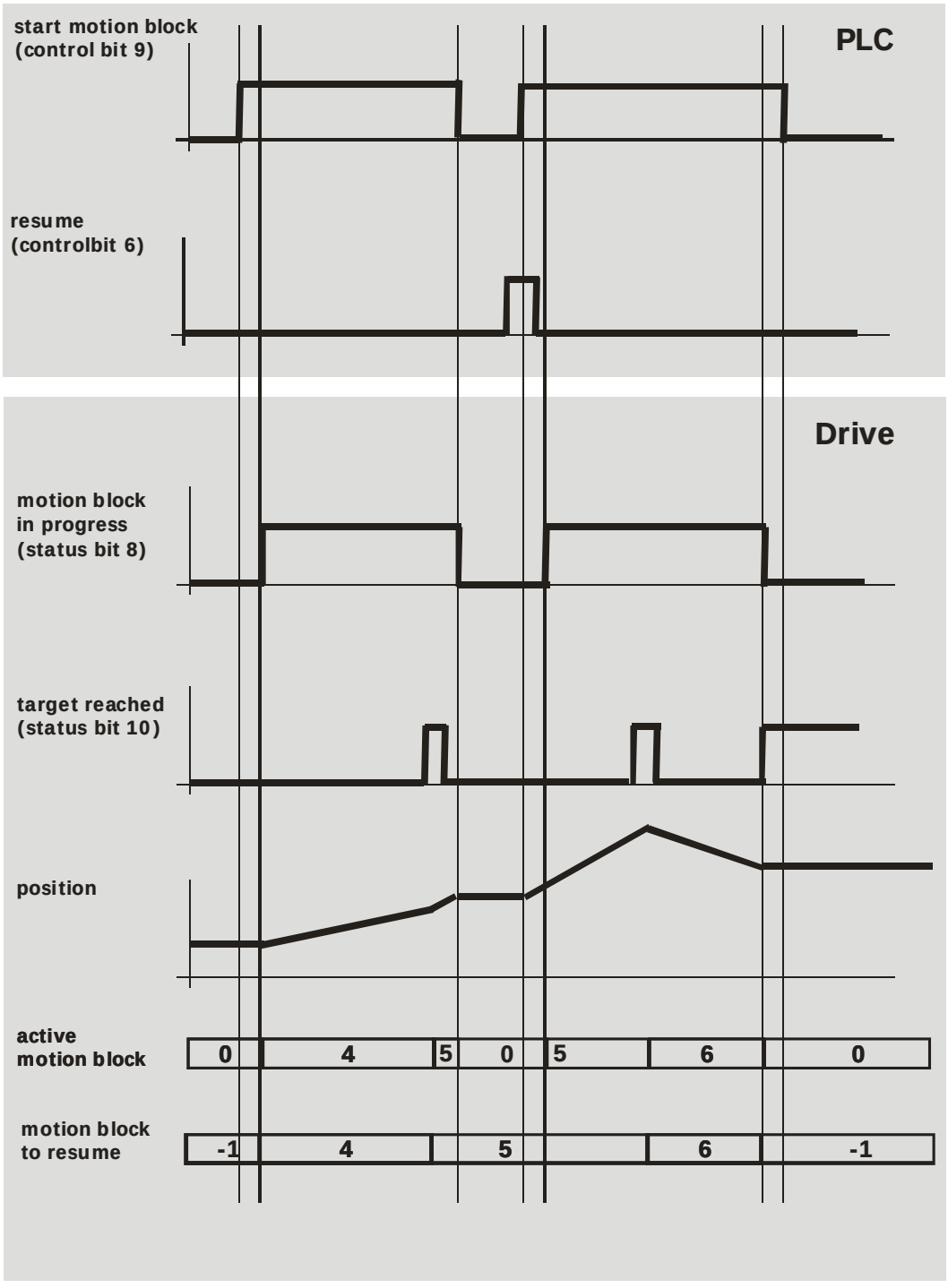
“single motion block” **sequence mode (control bit 4) = 0**
2 motion blocks 7 + 10



“ motion block sequence” sequence mode (control bit 4) = 1
sequence = motion block 4, 5, 6



“interrupted motion block sequence” **sequence mode (control bit 4) = 1**
sequence = motion block 4, 5, 6
motion block 5 interrupted



14.4.8.1 Example Sequence

To start the Table travel record mode, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000		Disable voltage
	Status word =		0x0050	Switch On Disabled
2	Modes of Operation =	-1		(Table travel record mode)
3	Control word =	0x0006		Shutdown
	Status word =		0x0031	Ready to switch on
4	Control word =	0x0007		Switch On
	Status word =		0x0033	Switched On
5a	Control word =	0x000F		Enable Operation.
	Status word =		0xnn37	Operation enabled
5b	Control word =	0x020F		Start Motion Block 1 as Single Motion Block.
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.
5c	Control word =	0x0A0F		Start Motion Block 2 as Single Motion Block.
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.
5d	Control word =	0x120F		Start Motion Block 3 as Single Motion Block.
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.
5e	Control word =	0x021F		Start Motion Block 1 in Sequence Mode
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.
5f	Control word =	0x004F		Resume last Motion Block as Single Motion Block
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.
5g	Control word =	0x005F		Resume last Motion Block in Sequence Mode
	Status word =		0xn337	Operation enabled and Positioning active.
	Status word =		0xn637	Operation enabled and Target reached.

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

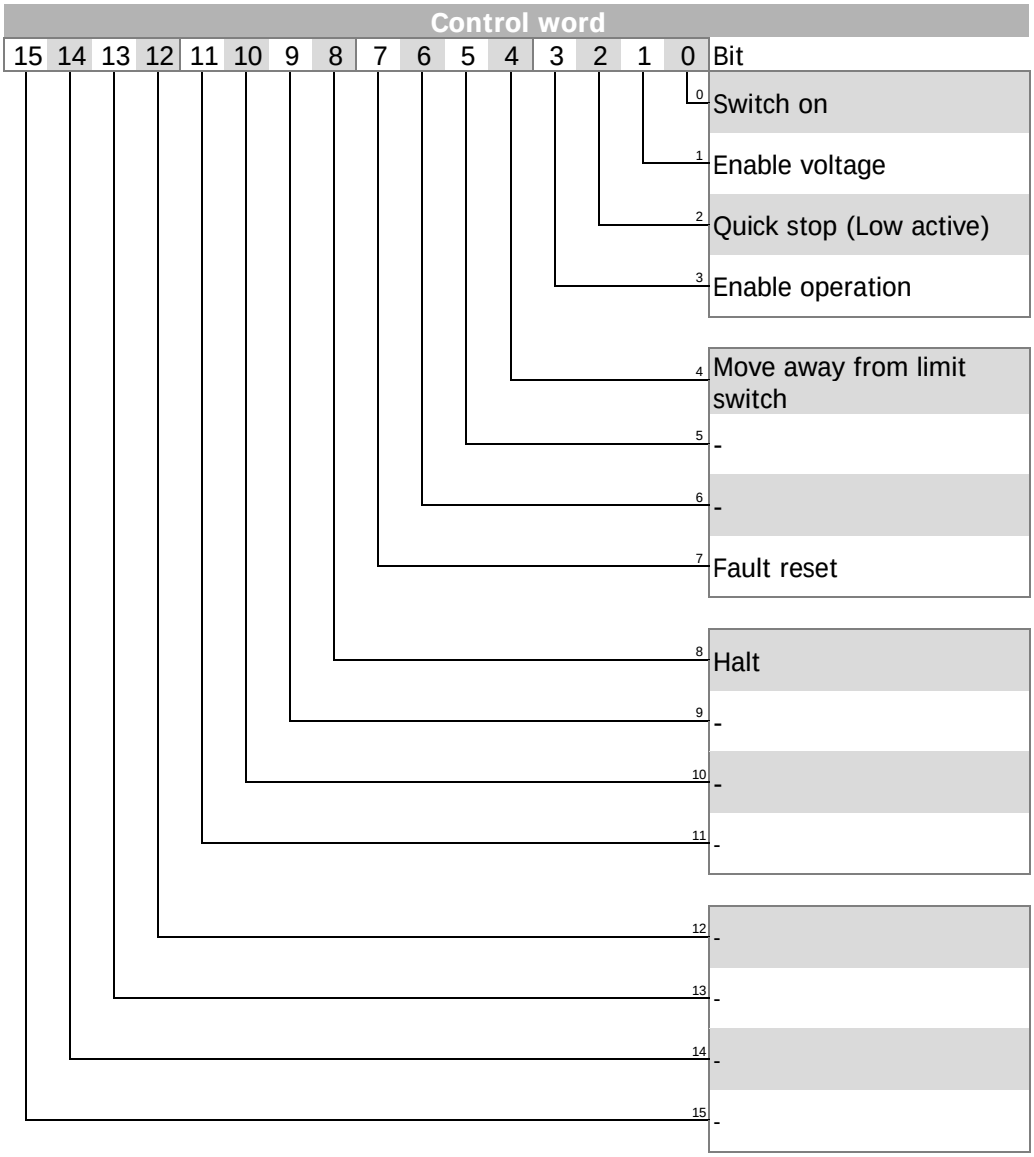
14.4.9 Move away from Limit switches

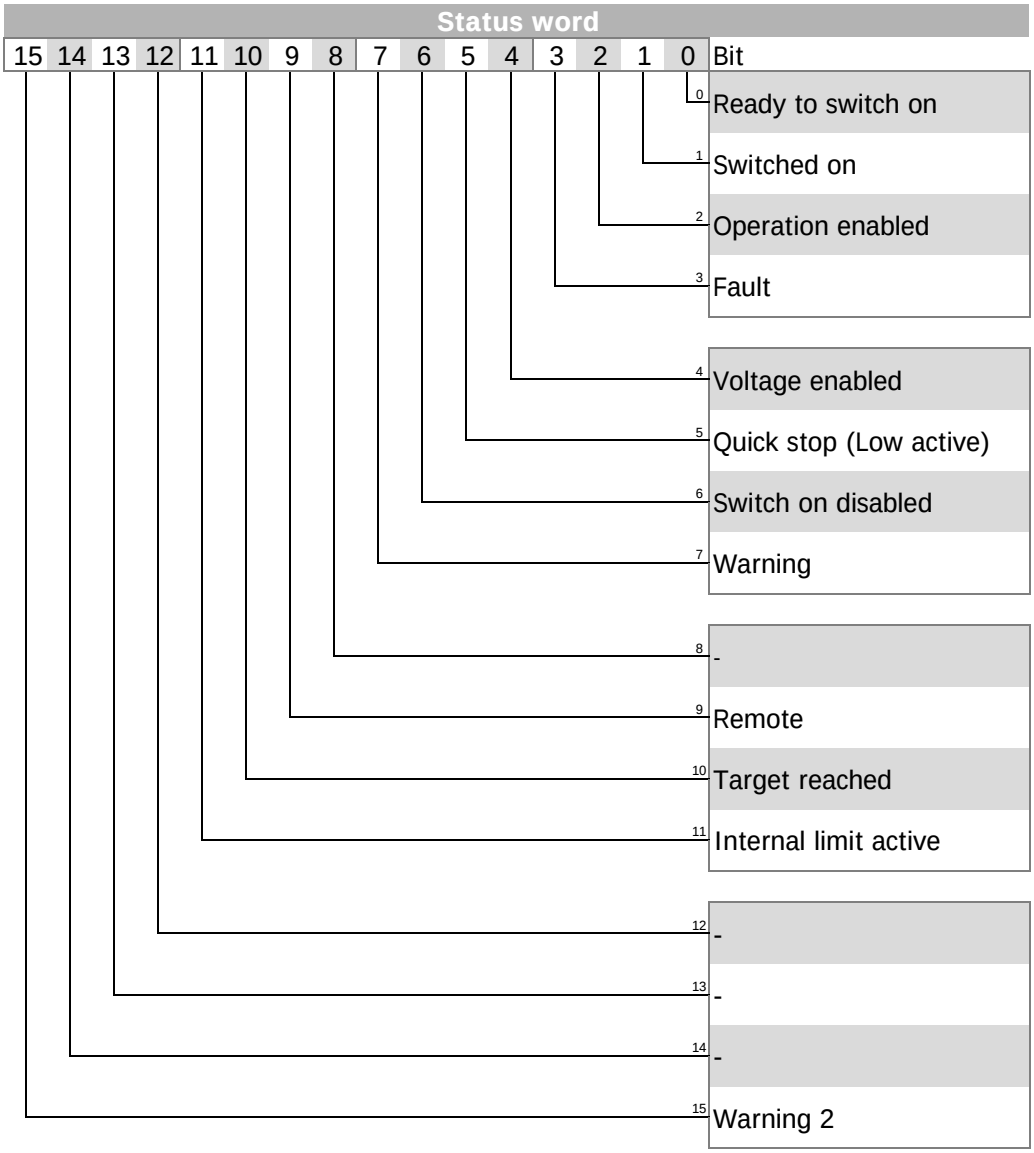
The Move away from Limit switches mode is selected via object 0x6060/0 *Modes of operation* = **0xFE** =**-2**.
In the Move away from Limit switches mode the drive moves independently from a triggered limit switch back into the valid Travel area.

Related objects:

0x6040	Control word	0x6085	Quick stop deceleration
0x6041	Status word	0x6099/2	Homing: Creep Speed
0x6046	Velocity min max amount	0x609A	Homing: Acceleration
0x6060	Modes of operation	0x6085	Quick stop deceleration
0x6061	Modes of operation display		

In table travel mode the "operation mode specific" and “manufacturer specific” bits of *Control word* and *Status word* are used as shown:





NOTE

The mode “Move away from limit switch” works under normal conditions with hardware limit switches. For Software limit switches the mode only works if a Software limit switch *Fault Reaction* **1144** with reaction “error” is selected. If a setting with warning (in example “10-Warning”) is selected, the software limit switch is not triggering a fault and therefore the mode “Move away from limit switch” won’t move away from the software limit switch.

NOTE

The mode “Move away from Limit Switch” must not be used after one of the following error messages occurred:

- F1444 Pos. SW Limit Switch < Neg. SW Limit Switch
- F1445 Pos. and Neg. HW-Limit Switch Simultaneously
- F1446 Limit Switch Incorrect Wired

If one of these faults has occurred, the wiring and parameterization must be checkd before the operation is continued.

Control word

Name	Value	Description
Move away from limit switch **Bit 4**	0	Don't start or cancel movement
	1	Start or resume movement from limit switch into travel area
Halt **Bit 8**	0	Execute instruction of bit 4 "Move away from limit switch"
	1	Stop axle with ramp of actual motion block, inverter remains in state "operation enabled"

Status word

Name	Value	Description	
Target reached **Bit 10**	0	Halt = 0:	Limit switch still active
		Halt = 1:	axle decelerates
	1	Halt = 0:	Limit switch relieved
		Halt = 1:	axle stopped (velocity 0)

Basic functions

In mode -2 „Move away from limit switch" the axle is automatically moved out of the range of a triggered hardware or software limit switch. The sense of rotation results from the active limit switch: if the positive limit switch was triggered, the drive is moved in negative direction and vice versa.

The Moving away procedure is started by Control Word bit 4 „Move away from limit switch" in state „operation enabled". The axle is accelerated to Object 0x6099 Homing speeds / Subindex 2 „Homing Speed 2 - search for zero" using the ramp defined by Object 0x609A Homing acceleration. As soon as the active software or hardware limit switch is released, the drive is stopped. After reaching zero speed, status-bit 10 „target reached" is set.

If both senses of rotation are blocked, in example after pos. and neg. hardware limit switch were triggered simultaneously, the error message "F1449 Both Directions Locked" is generated. In this case the function "Move away from limit switch" cannot be used.

NOTE

During the phase of moving away from the hardware limit switch the hysteresis defined by parameter 1149 is active. After detecting the edge of the limit switch the axle is moved for at least the defined hysteris distance.

Setting Halt to 1 interrupts the actual "Move away from Limit switch" movement. The axle is stopped. When reaching velocity 0 status bit "target reached" is set to 1. The drive remains in state operation enabled. Resetting Halt to 0 restarts processing of the "Move away from Limit switch" movement and status bit "target reached" is set to 0.

14.4.9.1 Example Sequence

To move away from the limit switch, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000	Disable voltage
	Status word =	0x0050	Switch On Disabled
2	Modes of Operation =	-2	(Move away from limit switch)
3	Control word =	0x0006	Shutdown
	Status word =	0x0031	Ready to switch on
4	Control word =	0x0007	Switch On
	Status word =	0x0033	Switched On
5	Control word =	0x000F	Enable Operation.
	Status word =	0xnn37	Operation enabled
6	Control word =	0x001F	Move away from limit switch.
	Status word =	0xn2B7	Operation enabled and Positioning active.
	Status word =	0xn637	Operation enabled and Limit switch no longer active (Target reached).

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

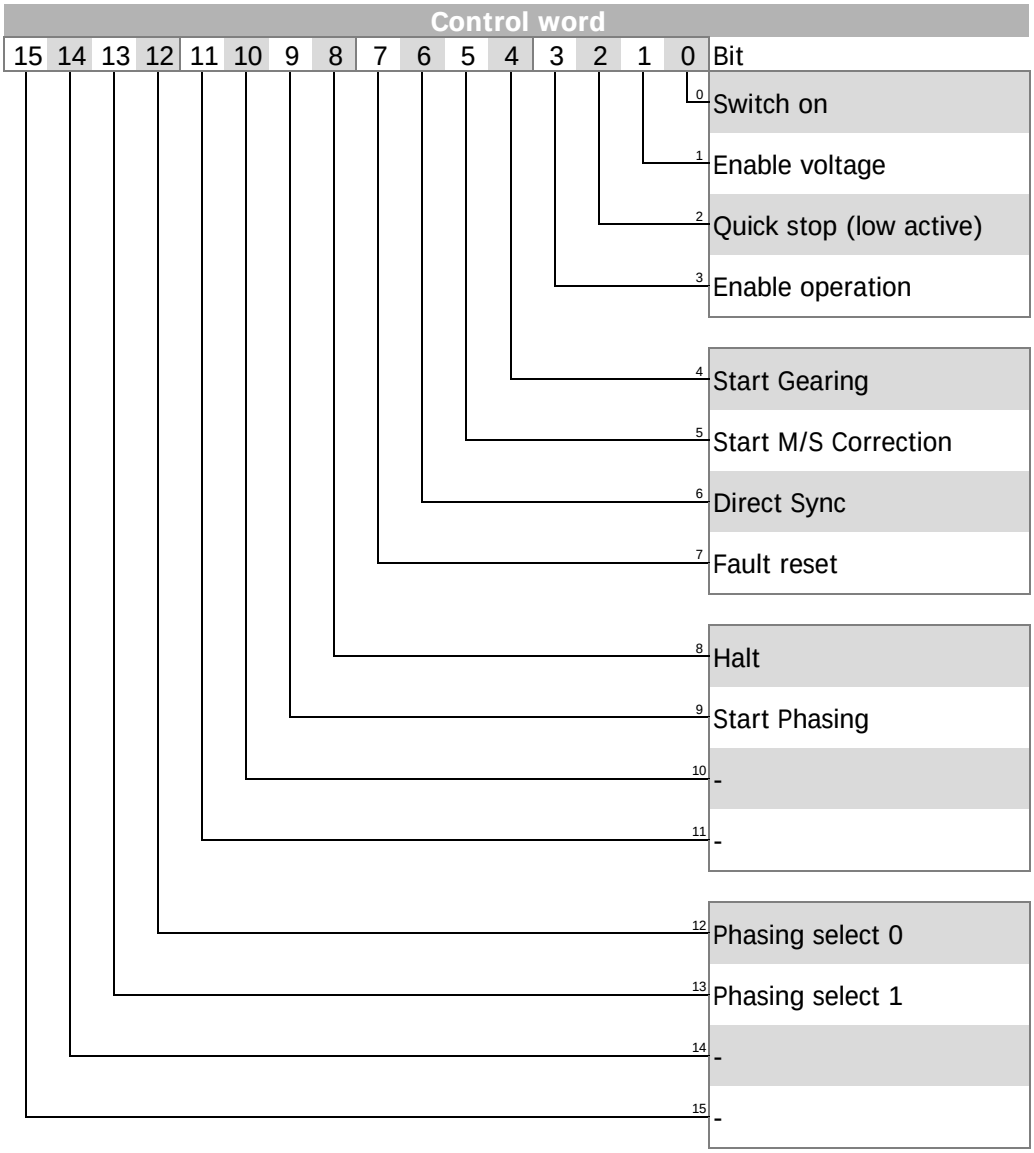
14.4.10 Electronic Gear: Slave

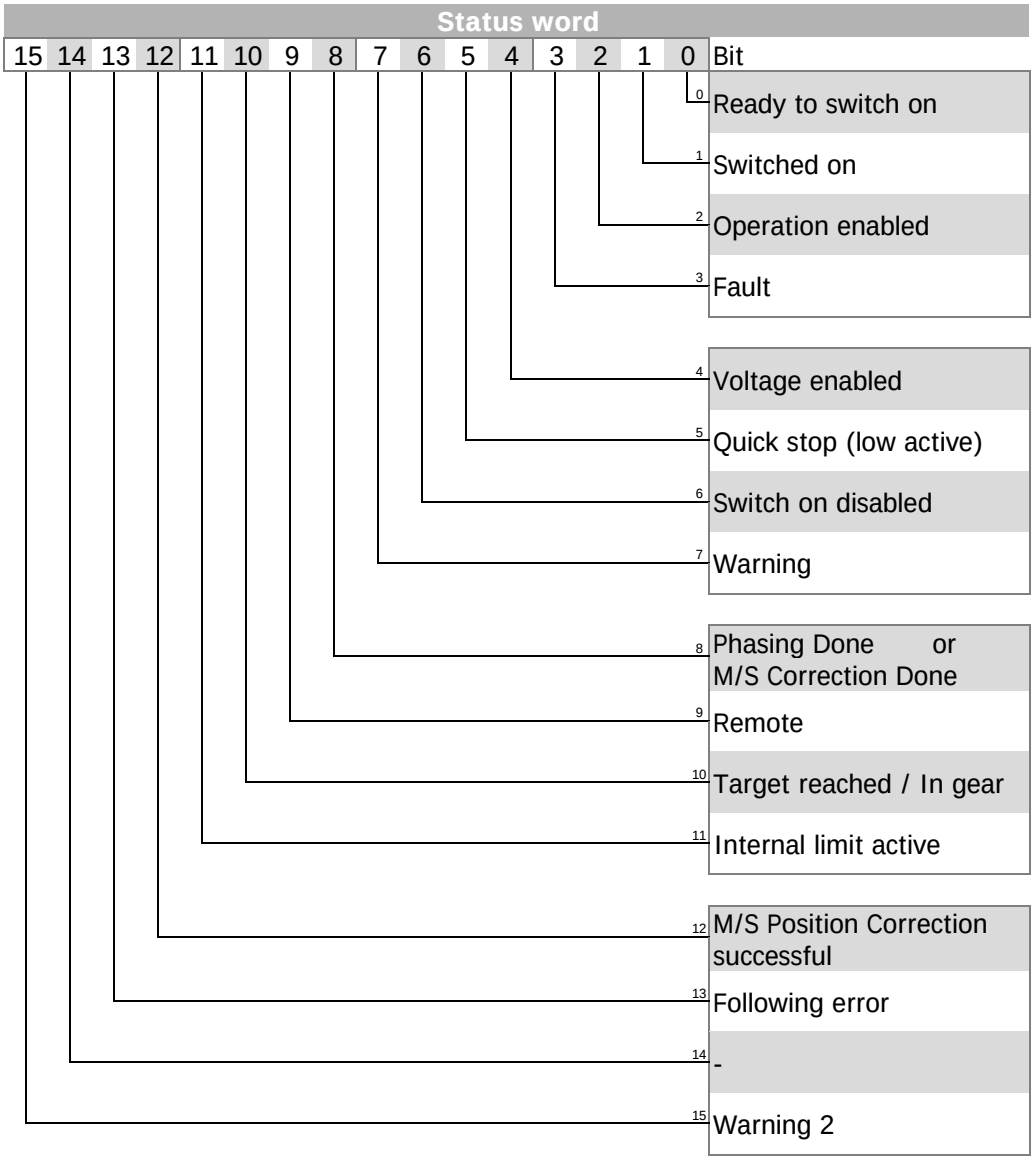
The *Electronic Gear: Slave* mode is selected via object 0x6060/0 *Modes of operation* = **0xFD** =**-3**. In *Electronic Gear: Slave* mode the drive follows as Slave drive a Master drive.

Related objects:

0x6040	Control word	0x6064	Position actual value
0x6041	Status word	0x6065	Following error window
0x6046	Velocity min max amount	0x6066	Following error time
0x6060	Modes of operation	0x6067	Position window
0x6061	Modes of operation display	0x6068	Position window time
0x5F10	Electronic Gear: Gear factor	0x6085	Quick stop deceleration
0x5F11	Electronic Gear: Phasing 1	0x5F18	M/S Synchronization offset
...	...		
0x5F14	Electronic Gear: Phasing 4		

In Electronic Gear: Slave mode the "operation mode specific" and “manufacturer specific” bits of *Control word* and *Status word* are used as shown:





⚠ WARNING

Dangerous state due to faulty parameterization

- The function Mater/Slave Position Correction is only allowed to be used after complete setup of this function. Check for parameter setup chapter 14.4.10.1 “Master/Slave Position Correction”.

Control word

Name	Value	Description
Start Gearing **Bit 4**	0	Stop axle with ramp 0x6084
	1	Start electronic gear with reference value master speed with ramp 0x6083
Start M/S Correction **Bit 5**	0	M/S Correnction not started.
	1	Start Master/Slave Postion correction. See chapter 14.4.10.1.
Direct Sync **Bit 6**	0	Direct Synchronisation enabled.
	1	Direct Synchronisation disabled.
Halt **Bit 8**	0	Execute instruction of bit 4 “Start Gearing”
	1	Stop axle with ramp of actual motion block, inverter remains in state “operation enabled”
Start Phasing **Bit 9**	0	Phasing disabled / aborted
	1	Start Phasing with profile defined by Bits 12 & 13
Phasing select 0...1 **Bit 12...13**	n	Phasing Profile= n + 1

Phasing select:

Control word															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Ph. Sel.				Pha	Halt		DS		SG				
		1	**0**												

Phasing Profile = motion block select + 1:

Phasing select		Phasing Profile
Bit 13	Bit 12	
0	0	1
1	1	2
0	0	3
1	1	4

Status word

Name	Value	Description
Phasing Done **Bit 8**	0	Phasing in progress or none started yet. or M/S correction in progress or none started yet.
	1	Phasing finished. or M/S correction finished.
Target reached / In gear **Bit 10**	0	Halt = 0: Electronic gear not coupled
		Halt = 1: axle decelerates
	1	Halt = 0: Electronic gear is coupled
		Halt = 1: axle has velocity 0
M/S Position Correction succesful **Bit 12**	0	M/S Korrektur is running or wasn’t started yet.
	1	M/S Korrektur finished. See chapter 14.4.10.1.
Following error **Bit 13**	0	No following error
	1	Following error

Basic functions

Mode "-3 Electronic Gear: Slave" implements a gearing operation for a slave drive to a master drive. The master of the Electronic gear must be connected to the Slave device via wire or Systembus (recommended). The master input in the Slave device is selected via *Master Position Source* **1122**.

Master Position Source 1122	Function
0 - Off	No source selected.
1 - Encoder 1	The current speed and position of the master drive is taken over from encoder input 1.
2 - Encoder 2 / Resolver	The current speed and position of the master drive is taken over from encoder input 2 or resolver.
11 - RxPDO1.Long1 extrapolated	The current position of the master drive is taken over by the process data channel RxPDO1.Long1 of the system bus. Additionally, the data received are extrapolated, even for slow settings of TxPDO Time of the master. Depending on the application, select a setting of the corresponding TxPDO.Long of the master: – "606 – Internal Act. Position (16/16)", mechanical position of master drive. Value **doesn't change** if the master makes a homing. – "607 – Act. Position (16/16)", mechanical position of master drive. Value **changes** if the master makes a homing. – "620 – motion profile gen.: Internal Ref. Position", reference position of master drive; advantage: improved controller properties. Value **doesn't change** if the master makes a homing. – "627 – motion profile gen.: Ref. Position", reference position of master drive; advantage: improved controller properties. Value **changes** if the master makes a homing. The settings 607 and 627 are only to be used in exceptional cases. In most applications sources 606 and 620 are more useful.

In setting "11 - RxPDO1.Long1 extrapolated" of parameter *Master Position Source* **1122** the system bus synchronization must be set to 1 or 10 to ensure a reliable function of *Operation Mode* **1180**.

Operation mode 1180
0 - Off [1]
1 - RxPDO1 [2]
2 - RxPDO2 [3]
3 - RxPDO3 [3]
10 - SYNC

[1] If the error message "F1453 Systembus-Synchronization not activated" is displayed when the slave drive is started, one of the operation modes 1, 2, 3 or 10 must be selected.

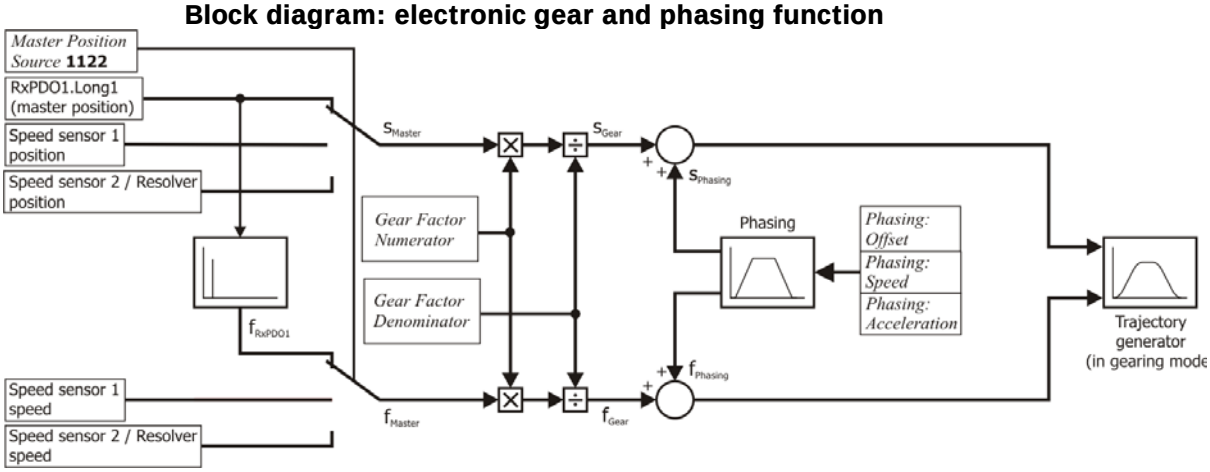
[2] Synchronization of processing with data telegram or cyclic sending of SYNC telegram.

[3] Not recommended for el. gear because no extrapolation done.

The synchronization of several drives needs high refresh rates to assure optimum results. Set the corresponding time (i.e. *TxPDO1 Time* **931**) at the transmit side to a low value. For the usage of the sync-function at the system bus set *SYNC-Time* **919** to a low value.
The bus load of the system bus must have sufficient reserves for proper operation.

The system bus is described in the manuals of the extension modules with system bus interface.

Block diagram: electronic gear and phasing function



The master position and speed is multiplied with the *gear factor*. When a *Phasing* is started, the Phasing profile is added to the Master Speed until the Phasing offset is reached.

The *gear factor* is defined by the following Objects or parameters:

Object	Parameter
0x5F10/1 Gear factor Numerator	**1123** *Gear Factor Numerator*
0x5F10/2 Gear factor Denominator	**1124** *Gear Factor Denominator*
0x5F10/3 Gear factor Resync on change	**1142** *Resync. on Change of Gear-Factor*

The *Phasing* is defined by the following Objects or parameters:

Object	Parameter
0x5F11/1 Phasing 1: Offset	**1125.1** *Phasing: Offset*
0x5F12/1 Phasing 2: Offset	**1125.2**
0x5F13/1 Phasing 3: Offset	**1125.3**
0x5F14/1 Phasing 4: Offset	**1125.4**
0x5F11/2 Phasing 1: Speed	**1126.1** *Phasing: Speed*
0x5F12/2 Phasing 2: Speed	**1126.2**
0x5F13/2 Phasing 3: Speed	**1126.3**
0x5F14/2 Phasing 4: Speed	**1126.4**
0x5F11/3 Phasing 1: Acceleration	**1127.1** *Phasing: Acceleration*
0x5F12/3 Phasing 2: Acceleration	**1127.2**
0x5F13/3 Phasing 3: Acceleration	**1127.3**
0x5F14/3 Phasing 4: Acceleration	**1127.4**

Start Electronic Gear and Status bits
Gearing operation is started by Control word bit 4 “Start Gearing”. The drive accelerates according to 0x6083 *Profile acceleration*. As soon as the slave speed is coupled to the master speed, Status word bit 10 “Target reached / Gear coupled” is set. The coupling conditions are set with objects 0x5F15 *In gear threshold* and 0x5F16 *In gear time*.

The coupled slave axis is controlled by the master. While the Slave is coupled speed, acceleration and deceleration is defined by the master. While coupled the Objects 0x6083 and 0x6084 have no effect.

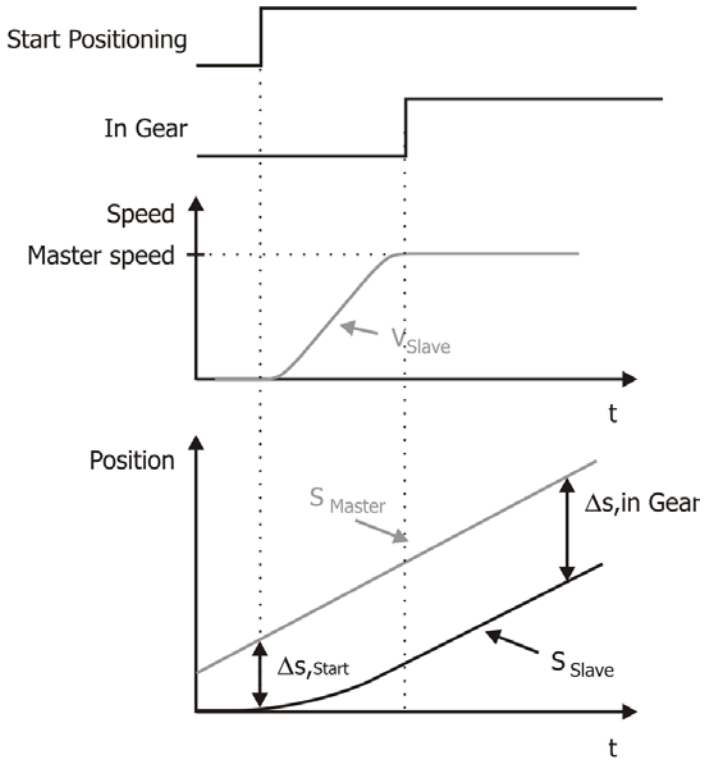
Setting *Halt* to 1 interrupts an actual processed movement. The axle is stopped with the ramp 0x6084. The “target reached” bit is reset at the start of the deceleration and set again after reaching zero speed. The drive remains in state *operation enabled*. Resetting *Halt* to 0 restarts processing of the interrupted movement. The “target reached” bit is reset at the start of the acceleration and set again when meeting the “In gear” criteria set in objects 0x5F15 *In gear threshold* and 0x5F16 *In gear time*..

Phasing
With the phasing function, the slave position is offset from the received position of the master by the Phasing-Position.

The Phasing is described in chapter 12.4.19 “0x5F11/n...0x5F14/n Phasing 1...4”.

Direct Synchronization

Function without Direct Synchronization
(“Standard Synchronization”)
The drive accelerates the master speed at the ramps parameterized in the motion block. As soon as the master speed is reached for the first time, the drive is synchronized with the master drive. The slave is engaged at the current position and operates at a synchronous angle with the master. In the case of a relative positioning operation, this engaging position is used as the start position.
The relative position shift between master and slave due to the acceleration is not compensated.

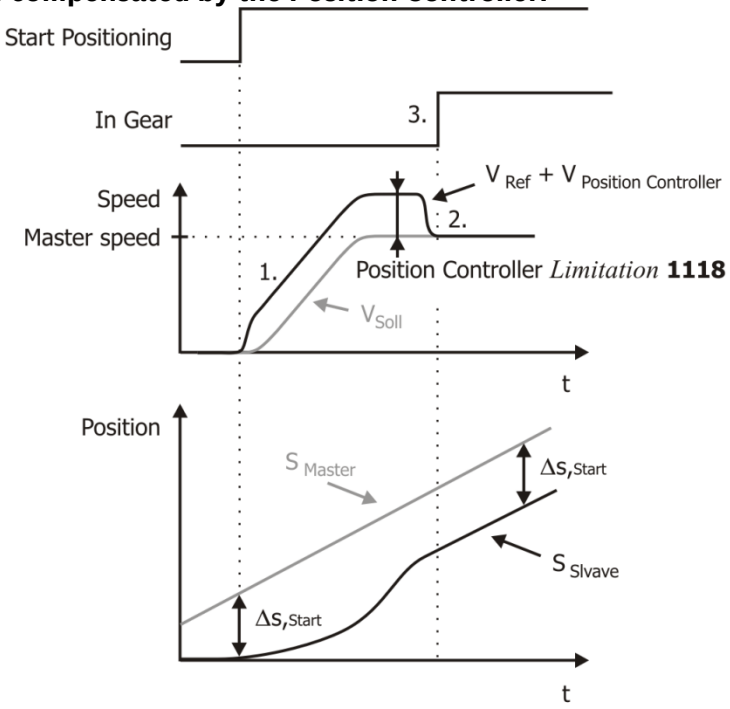


Function with Direct Synchronization

The drive accelerates the master speed at the ramps parameterized in the motion block. At the start of a motion block the drive is sychronised with the master drive directly. The master speed is processed by the position controller directly.

The acceleration and deceleration for synchronization occurs according to the characteristic of an S-shaped curve.

The relative position shift between master and slave due to the acceleration is compensated by the Position Controller.



14.4.10.1 Master/Slave Position Correction

NOTE

When using this functionality master drive and slave drive have to use the same mechanical characteristics (i.e. gear transmission ratios) and use the same reference system.

Preparations Master drive

The Master drive must be set up as follows:
TxPDO2 Identifier **927 =** 640 (or a different not used Identifier)
TxPDO2 Function **932** = 1 – controlled by time or 2 – controlled by SYNC
TxPDO2.Long1 **964** = 743 – Act. Position [User Units]

Additionally the following parameters must be set according to the electronic gear:
TxPDO1.Long1 **954** corresponding to the description of *Source Master position* **1122**
TxPDO1 Identifier **925 =** 384 (or a different not used Identifier)
TxPDO1 Function **930** = 1 – controlled by time or 2 – controlled by SYNC

Preparations Slave drive

The Slave drive must be set up as follows:
RxPDO2 Function **926 =** 640 (or the Identifier defined in the Master drive)

Additionally the following parameters must be set according to the electronic gear:
RxPDO1 Function **924 =** 384 (or the Identifier defined in the Master drive)
Source Master position **1122** = 11 – RxPDO1.Long

The function Master/Slave Position Correction expects the Target Position [u] always in RxPD2.Long. When using this function RxPDO2.Long1 and also RxPDO2.Word1, RxPDO2.Word2, RxPDO2.Boolean1 and RxPDO2.Boolean2 are not allowed to be used for any other purpose.

Starting of Master/Slave Position Correction in Slave drive

To start the Master/Slave Position correction at first Bit 4 and then Bit 5 have to be set in the Control word. Bit 5 is only allowed to be set when Bit 10 In Gear is shown in the Status word.

By setting Bit 5 in the Control word the Slave drive is started to position to the Master position + Offset.

The acceleration is done with the object 0x609A/0 0Homing acceleration (o Parameter *Acceleration* **1134**). The used velocity can be set up via 0x6099/1 Homing speed (or Parameter *Fast Speed* **1132**).

As long as the Master/Slave Position correction is executed, Bit 12 is deactivated in the Status word. When the Master/Slave Position correction was finished successfully Bit 12 is set.

During the Correction sequence the Status word bit 8 “Master/Slave Position correction” is set to “Low”. As soon as the Master/Slave Position correction is finished or cancelled, the Bit is set to “High”. After first switch-on (or after a device reset) the “Phasing Done” bit is also “Low”.

Since Bit 8 is also used for Phasing, always the last started function is signalled by this bit.

Offset Reference

The Offset for the M/S Synchronization can be set via Object 0x5F18/0.

Object	Parameter
0x5F18/0 M/S Synchronization offset	1284 *M/S Synchronization offset*

Application limitations

The function can be used in most of all applications without any limitations. In applications with very long travelling distances the following must be checked:

- The position difference to be compensated must not be greater than 2^{15}-1 motor revolutions.
- The position difference to be compensated must not be greater than 2^{31}-1 user units.

Depending on the used reference system it can vary, which limit is decisive. Always the smaller limit must be complied with.

A motor with a reference speed of 6000 rpm would have to travel for around 5.5 minutes into one direction to exceed this limit.

14.4.10.2 Example Sequence

To start the Electronic Gear: Slave mode, the correct sequence has to be sent from the PLC.

1	Control word =	0x0000	Disable voltage
	Status word =	0x0050	Switch On Disabled
2	Modes of Operation =	-3	(Electronic Gear: Slave mode)
3	Control word =	0x0006	Shutdown
	Status word =	0x0031	Ready to switch on
4	Control word =	0x0007	Switch On
	Status word =	0x0033	Switched On
5	Control word =	0xn327	Enable Operation.
	Status word =	0xn337	Operation enabled
6a	Control word =	0xn727	Start Electronic Gear without Direct Synchronization
	Status word =	0xn327	Operation enabled, Slave not (yet) coupled, Phasing not finished.
	Status word =	0xn337	Operation enabled, Slave not (yet) coupled, Phasing finished.
	Status word =	0xn727	Operation enabled, Slave coupled, Phasing not finished.
	Status word =	0xn737	Operation enabled, Slave coupled, Phasing finished.
6b	Control word =	0x005F	Start Electronic Gear with Direct Synchronization
	Status word =	See 6a	See 6a
7a	Control word =	0x021F	Start Electronic Gear without Direct Synchronization and Phasing Profile 1
	Status word =	See 6a	See 6a
7b	Control word =	0x121F	Start Electronic Gear without Direct Synchronization and Phasing Profile 2
	Status word =	See 6a	See 6a
7c	Control word =	0x221F	Start Electronic Gear without Direct Synchronization and Phasing Profile 3
	Status word =	See 6a	See 6a
7d	Control word =	0x321F	Start Electronic Gear without Direct Synchronization and Phasing Profile 4
	Status word =	See 6a	See 6a
8a	Control word =	0x025F	Start Electronic Gear with Direct Synchronization and Phasing Profile 1
	Status word =	See 6a	See 6a
8b	Control word =	0x125F	Start Electronic Gear with Direct Synchronization and Phasing Profile 1
	Status word =	See 6a	See 6a
8c	Control word =	0x225F	Start Electronic Gear with Direct Synchronization and Phasing Profile 3
	Status word =	See 6a	See 6a
8b	Control word =	0x325F	Start Electronic Gear with Direct Synchronization and Phasing Profile 4
	Status word =	See 6a	See 6a
9	Control word =	0x001F 0x003F	Enable Operation, the Slave drive synchronizes to the Master position.
	Status word =	0xnn37 0x1n37	Operation enabled M/S Position Correction finished.

⚠ WARNING

Dangerous state due to new mode!

- When 0x6060 *Modes of Operation* is changed during operation (Control word = 0xnnnF), a dangerous state can occur in the new mode.
- Checking the status word before changing 0x6060 *Modes of Operation* (i.e. check state 0xnn33).

After the sequence of the first four Control word s was processed correctly, the ACU is enabled (dark marked table area).
In “Operation enabled” state (0xnnnF) the Motion Control states can be changed (white marked area in table).
With the control word transition from 0xnnnF to 0x0007 the velocity mode is stopped. After that it is possible to start again with 0xnnnF.
While 0x0007 is active, it is also possible to change the modes of operation without any danger. After changing 0x6060 *modes of operation* to another value you can start the new operation mode with the according sequence.

Bit 5 “Start Position Correction” is only allowed to be used when the Slave is in gear (Status word Bit 10).
Bit 5 “Start Position Correction” should be used for optimum results when the master drive doesn’t move.
When Bit 5 of the Control word is reset to “0” the movement is interrupted.

15 Parameter list

The parameter list is structured according to the menu branches of the operating unit. For better clarity, the parameters have been marked with pictograms:

The parameter is available in the four data sets.

☑ The parameter value is set by the SET-UP routine.

⊗ This parameter cannot be written when the frequency inverter is in operation.

15.1 Actual values

No.	Description	Unit	Display range	Chapter
	Actual values of the frequency inverter			
228	Internal Reference Frequency	Hz	-1000.00 ... 1000.00	14.3.3
249	Active Data Set	-	1 ... 4	14
260	Current error	-	0 ... 0xFFFF	12.5.2 16.4
270	Warnings	-	0 ... 0xFFFF	16.2
274	Application Warnings	-	0 ... 0xFFFF	16.3
282	Reference Bus Frequency	Hz	-1000.00 ... 1000.00	14.3.3
283	Reference Ramp Frequency	Hz	-1000.00 ... 1000.00	14.3.3
1290	Node State (NMT)	-	0 ... 127	11.7
1291	CAN State (physical layer)	-	0 ... 4	6
1453	OS SyncSource Act	-	Selection	11.10
	Actual values of the frequency inverter			
1108	Actual Position	u	--2147483647 ... 2147483647	12.5.14
1109	Act. Contouring Error	u	--2147483647 ... 2147483647	12.5.41

The parameters *Current error* **260**, *Warnings* **270** and *Application Warnings* **274** are only accessible via the communication channel of objects PPO1 and PPO2. It cannot be accessed via the VPlus program or the KP500 control unit.

15.2 Parameters

No.	Description	Unit	Display range	Factory setting	Chapter
	CAN bus				
276	CAN Interface	-	Selection	1 - CM-CAN	9
385	CAN Baud Rate	-	Selection	6 - 250 kBit/s	7
387	CAN Node Number	-	-1 ... 127	-1	8
388	Bus Error Behaviour	-	Selection	1 - Error	10, 12.5.1
	Rated motor parameters				
373	No. of Pole Pairs	-	1 ... 24	2	12.5
	Bus control				
392	State-transition 5	-	Selection	2 - Ramp	14.3.2
412	Local/Remote	-	Selection	44 - Ctrl. Cont.+KP, Dir. Cont.+KP	14
	Data set change-over				
414	Data set selection	-	0 ... 4	0	14
	Frequency ramps				
420	Acceleration (Clockwise)	Hz/s	0.01 ... 9999.99	5.00	12.5.9
421	Deceleration (Clockwise)	Hz/s	0.01 ... 9999.99	5.00	12.5.10, 12.5.11

No.	Description	Unit	Display range	Factory setting	Chapter
422	Acceleration (Anticlockwise)	Hz/s	-0.01 ... 9999.99	-0.01	12.5.9
423	Deceleration (Anticlockwise)	Hz/s	-0.01 ... 9999.99	-0.01	12.5.10, 12.5.11
424	Emergency stop (Clockwise)	Hz/s	0.01 ... 9999.99	5.00	12.5.11, 14.3.1
425	Emergency stop (Anticlockwise)	Hz/s	0.01 ... 9999.99	5.00	12.5.11, 14.3.1
434	Ramp Setpoint	-	Selection	3 - Internal + Line Setpoint	14.3.3
Digital outputs					
549	Max. Control Deviation	%	0.01 ... 20.00	5.00	14.1, 14.2
Stopping behavior					
637	Switch-Off Threshold	%	0.0 ... 100.0	1.0	14.3.1, 14.3.2
638	Holding Time	s	0.0 ... 200.0	1.0	14.3.1, 14.3.2
Electronic Gear					
1122	Master Position Source	-	Selection	0-Off	14.4.10
Systembus					
1180	Synchronization	-	Selection	0-Off	14.4.10
Master/Slave Position Correction					
1284	Master/Slave Synchonization Offset	-	Selection	0 u	14.4.10.1
Motion Control Interface					
1292	S. Modes of Operation	-	Selection	801 – 0x6060	13.6
1293	S. Target Position	-	Selection	802 – 0x607A	13.6
1294	S. Profile Velocity	-	Selection	803 – 0x6081	13.6
1295	S. Acceleration	-	Selection	804 – 0x6083	13.6
1296	S. Deceleration	-	Selection	805 – 0x6084	13.6
1297	S. Reference Speed vl [rpm]	-	Selection	806 – 0x6042	13.6
1299	S. Special Function Generator	-	Selection	9 - Zero	13.6
1297	S. Target Velocity	-	Selection	806 – 0x6042	13.6
CANopen® Mux/DeMux					
1420	CANopen Mux Input Index (write)[1)]	-	EEPROM 0...16 RAM 17... 33	1	12.4.5
1421	CANopen Mux Input Index (read) [1)]	-	EEPROM 0...16 RAM 17... 33	1	12.4.5
1422	CANopen Mux Input	-	Selection	7 - Off	12.4.5
1423	CANopen Percentage Actual Value Source	-	Selection	52 - Analog Input MFI1A	12.4.8
1451	CANopen OS Synctime	-	700...900 us	800 us	11.10
1452	OS_SyncSource	-	Selection	52 - Analog Input MFI1A	11.10
Motion Control Override					
1454	Override Modes of Operation	-	Selection	0	13.7
1455	Override Target Position	-		-1 u	13.7
1456	Override Profile Velocity	-		-1 u/s	13.7
1457	Override Profile Acceleration	-		-1 u/s²	13.7
1458	Override Profile Deceleration	-		-1 u/s²	13.7
1459	Override Target velocity vl [rpm]	-		-1 rpm	13.7
1460	Override Target velocity pv [u/s]	-		-1 u/s	13.7

1)	Non volatile (fixed Parameterization)		Volatile	
	0:	All indexes in EEPROM	17:	Alle indexes in RAM
	1...16:	One Index in EEPROM	18...33:	One Index 1...16 in RAM

Further parameters are described in the Operating instructions and the Application manual “Positioning”.

The setting “0” for *CANopen Mux Input Index (write)* **1420** changes all Data in EEPROM and. RAM.

The parameter *Data set selection* **414** is only accessible via the manufacturer objects 0x2nnn. It cannot be accessed via the VPlus program or the KP500 control unit.

Further parameters are described in the Operating instructions and the Application manual “Positioning”.

16 Annex

In the Annex different overviews and Tools are displayed for the usage of the communication interface.

16.1 Control word/ Status word overview

16.1.1 Control Word overview (without Sync Modes)

The tables on this page list in an overview the functionality of the **Control Word** bits.

Bit	*Standard (No Positioning)*	*Positioning without MCI*	*MCI: Velocity Mode*	*MCI: Profile Velocity Mode*	*MCI: Profile Position Mode*
0	Switch On	Switch On	Switch On	Switch On	Switch On
1	Enable Voltage	Enable Voltage	Enable Voltage	Enable Voltage	Enable Voltage
2	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)
3	Enable Operation	Enable Operation	Enable Operation	Enable Operation	Enable Operation
4			Rfg enable		New setpoint
5			Rfg unlock		Change set immediately
6			Rfg use ref		Abs/rel
7	Fault reset	Fault reset	Fault reset	Fault reset	Fault reset
8	Halt	Halt	Halt	Halt	Halt
9					Change on setpoint
10					
11					
12					
13					
14					
15					

Bit	*MCI: Interpol. Position Mode*	*MCI: Homing Mode*	*MCI: Table travel record Mode*	*MCI: Move away from Limit Sw.*	*MCI: Electronic Gear: Slave*
0	Switch On	Switch On	Switch On	Switch On	Switch On
1	Enable Voltage	Enable Voltage	Enable Voltage	Enable Voltage	Enable Voltage
2	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)	Quick Stop (Low active)
3	Enable Operation	Enable Operation	Enable Operation	Enable Operation	Enable Operation
4	Enable ip-mode	Homing op. start	Sequence mode	Move away from LS	Start Gearing
5					
6			Resume		Direct Sync
7	Fault reset	Fault reset	Fault reset	Fault reset	Fault reset
8	Halt	Halt	Halt	Halt	Halt
9			Start motion block		Start Phasing
10					
11			Motion Block Select 0		
12			Motion Block Select 1		Phasing Profile Sel. 1
13			Motion Block Select 2		Phasing Profile Sel. 2
14			Motion Block Select 3		
15			Motion Block Select 4		

16.1.2 Status Word overview (without Sync modes)

The tables on this page list in an overview the funcionality of the **Status Word** bits.

Bit	*Standard (No Positioning)*	*Positioning without MCI*	*MCI: Velocity Mode*	*MCI: Profile Velocity Mode*	*MCI: Profile Position Mode*
0	Ready to Switch On	Ready to Switch On	Ready to Switch On	Ready to Switch On	Ready to Switch On
1	Switched On	Switched On	Switched On	Switched On	Switched On
2	Operation enabled	Operation enabled	Operation enabled	Operation enabled	Operation enabled
3	Fault	Fault	Fault	Fault	Fault
4	Voltage enabled	Voltage enabled	Voltage enabled	Voltage enabled	Voltage enabled
5	Quick Stop	Quick Stop	Quick Stop	Quick Stop	Quick Stop
6	Switch On Disabled	Switch On Disabled	Switch On Disabled	Switch On Disabled	Switch On Disabled
7	Warning	Warning	Warning	Warning	Warning
8		Homing done			
9	Remote	Remote	Remote	Remote	Remote
10	Target reached	Target reached	Target reached	Target reached	Target reached
11	Internal limit active	Internal limit active	Internal limit active	Internal limit active	Internal limit active
12				Speed	Set-point acknowl.
13				Max slippage error	Following error
14		Target Pos. reached			
15	Warning 2	Warning 2	Warning 2	Warning 2	Warning 2

Bit	*MCI: Interpol. Position Mode*	*MCI: Homing Mode*	*MCI: Electronic Gear: Slave*	*MCI: Move away from Limit Sw.*	*MCI: Electronic Gear: Slave*
0	Ready to Switch On	Ready to Switch On	Ready to Switch On	Ready to Switch On	Ready to Switch On
1	Switched On	Switched On	Switched On	Switched On	Switched On
2	Operation enabled	Operation enabled	Operation enabled	Operation enabled	Operation enabled
3	Fault	Fault	Fault	Fault	Fault
4	Voltage enabled	Voltage enabled	Voltage enabled	Voltage enabled	Voltage enabled
5	Quick Stop	Quick Stop	Quick Stop	Quick Stop	Quick Stop
6	Switch On Disabled	Switch On Disabled	Switch On Disabled	Switch On Disabled	Switch On Disabled
7	Warning	Warning	Warning	Warning	Warning
8			Motion Block in Progress		Phasing Done
9	Remote	Remote	Remote	Remote	Remote
10	Target reached	Target reached	Target reached	Target reached	Target reached
11	Internal limit active	Internal limit active	Internal limit active	Internal limit active	Internal limit active
12	IP-mode active	Homing attained	In gear		
13		Homing error	Following error		Following error
14					
15	Warning 2	Warning 2	Warning 2	Warning 2	Warning 2

16.1.3 Control Word overview for Sync Modes

The table on this page list in an overview the functionality of the **Control Word** bits.

Bit	*MCI: Sync Position Mode*	*MCI: Sync Velocity Mode*
0	Switch On	Switch On
1	Enable Voltage	Enable Voltage
2	Quick Stop (Low active)	Quick Stop (Low active)
3	Enable Operation	Enable Operation
4		
5		
6		
7	Fault reset	Fault reset
8	Halt	Halt
9		
10		
11		
12		
13		
14		
15		

16.1.4 Status Word overview for Sync modes

The table on this page list in an overview the functionality of the **Status Word** bits..

Bit	*MCI: Sync Position Mode*	*MCI: Sync Velocity Mode*
0	Ready to Switch On	Ready to Switch On
1	Switched On	Switched On
2	Operation enabled	Operation enabled
3	Fault	Fault
4	Voltage enabled	Voltage enabled
5	Quick Stop	Quick Stop
6	Switch On Disabled	Switch On Disabled
7	Warning	Warning
8		
9	Remote	Remote
10		
11		
12	Target Position ignored	Target velocity ignored
13	Following error	
14		
15	Warning 2	Warning 2

16.2 Warning messages

The various control functions and methods as well as the hardware of the frequency inverter contain functions that continuously monitor the application. In addition to the messages documented in the manual, the following warning messages are activated by the communication module.
The warning messages are given via parameter *Warnings* **270**, bit-coded according to the following scheme. The parameter *Warnings* **270** is meant to be read out by PLCs, parameter *Warnings* **269** shows the same information with a short text description in VPlus and Keypad KP500.

Warning messages		
Bit no.	**Warning code**	**Meaning**
0	0x0001	Warning Ixt
1	0x0002	Warning Short Term - Ixt
2	0x0004	Warning Long Term - Ixt
3	0x0008	Warning Heat sink Temperature Tc
4	0x0010	Warning Inside Temperature Ti
5	0x0020	Warning Limit
6	0x0040	Warning Init
7	0x0080	Warning Motor Temperature
8	0x0100	Warning Mains Failure
9	0x0200	Warning Motor Protective Switch
10	0x0400	Warning Fmax
11	0x0800	Warning Analog Input MFI1A
12	0x1000	Warning Analog Input A2
13	0x2000	Warning System bus
14	0x4000	Warning Udc
15	0x8000	Warning *Warning status application* **367**

The meaning of the individual warnings is described in detail in the operating instructions.

16.3 Warning Messages Application

When the highest bit in the Warning messages is set, a “Warning Message Application” is present. The Application warning messages are given via parameter *Application Warnings* **274**, bit-coded according to the following scheme.
Parameter *Application Warnings* **273** shows the warnings in clear text on the operator panel and the PC software tool VPlus.
Use Parameter *Application Warnings* **274** to access the Application warning codes via Field bus.

Warning Messages			
Bit-No.	**Bit-No.**	**Bit-No.**	
0	0x0001	BELT	- Keilriemen
1	0x0002	SW-LIM CW	– SW limit switch Clockwise
2	0x0004	SW-LIM CCW	– SW limit switch Counterclockwise
3	0x0008	HW-LIM CW	– HW limit switch Clockwise
4	0x0010	HW-LIM CCW	– HW limit switch Counterclockwise
5	0x0020	CONT	– Contouring error
6	0x0040	ENC	– Warning Absolute encoder
7	0x0080	User 1	– User Warning 1
8	0x0100	User 2	– User Warning 2
9	0x0200	(reserviert)	
10	0x0400	(reserviert)	
11	0x0800	(reserviert)	
12	0x1000	(reserviert)	
13	0x2000	(reserviert)	
14	0x4000	(reserviert)	
15	0x8000	(reserviert)	

The warnings are described in detail in the operating instructions or the application manual “Positioning” respectively.
The Warning Bit 6 “Absolute encoder” can be read out via Parameter **1274** in VPlus or **1273** via field bus. The Absolute encoder warnings are described in detail in the Extension manual EM-ABS-01.

16.4 Fault messages

The fault code that is stored after a fault occurs is made up of the fault group FXX (high Byte, hexadecimal) followed by the code number XX (low Byte, hexadecimal).

	Communication error		
	Code		**Meaning**
Motion Control Interface	F04	04	Control Deviation Position Controller
	F14	42	Pos. SW-Limit Switch
		43	Neg. SW-Limit Switch
		44	Pos. SW-Lim. Switch < Neg. SW-Lim. Switch
		45	Pos. and Neg. HW-Lim Switch Simultaneously
		46	Limit Switch Incorrect Wired!
		47	Pos. HW Limit Switch
		48	Neg. HW Limit Switch
		51	Clockwise Operation Locked
		52	Anti-Clockwise Operation Locked
		53	System bus-Synchronization not activated
		60	Pos. HW-Lim. Switch: Illegal Signal Source
		61	Pos. HW-Lim. Switch: Input disabled by PWM-/FF-Input
		62	Pos. HW-Lim. Switch: Input disabled by Index-Contr.
		63	Pos. HW-Lim. Switch: Wrong Op.-Mode for MFI1
		64	Pos. HW-Lim. Switch: Input disabled by Encoder 1
		65	Pos. HW-Lim. Switch: Input disabled by Encoder 2
		66	Pos. HW-Lim. Switch: Wrong Op.-Mode for EM-S1IOD
		70	Neg. HW-Lim. Switch: Illegal Signal Source
		71	Neg. HW-Lim. Switch: Input disabled by PWM-/FF-Input
		72	Neg. HW-Lim. Switch: Input disabled by Index-Contr.
		73	Neg. HW-Lim. Switch: Wrong Op.-Mode for MFI1
		74	Neg. HW-Lim. Switch: Input disabled by Encoder 1
		75	Neg. HW-Lim. Switch: Input disabled by Encoder 2
		76	Neg. HW-Lim. Switch: Wrong Op.-Mode for EM-S1IOD
	F15	xx	User-Defined Error in Motion Block xx (1 £ xx £ 32)
		70	No Homing Done
		71	Homing : Encoder-Mode w.o. Z-Impulse
		72	Both Directions Locked
		73	No Touch Probe Signal Detected
		74	M/S Position Correction: Master Position source not set
CANopen	F20	21	Bus OFF
		22	Guarding failure
		23	Error state
		24	SYNC error (SYNC timing)
		25	NMT state change (operational → xxx)
		26	RxPDO1 length error — (number of received bytes different to mapping)
		27	RxPDO2 length error
		28	RxPDO3 length error
		2A	RxPDO1 Timeout — (The RxPDO was not received in the set time. Check object 0x140n/5 Event time.)
		2B	RxPDO2 Timeout
		2C	RxPDO3 Timeout
	F23	nn	Heartbeat failure – nn = node address of the failed subscriber (hex)

The Actual error message can be read out by parameter access via parameter *Actual Error* **260** and via the Emergency Message or Object 0x1014.
Parameter *Actual Error* **259** shows the actual error in clear text on the operator panel and the PC software tool VPlus.
In addition to the fault messages stated, there are further fault messages described in the Operating Instructions. The faults of the Motion Control Interface (F14xx, F15xx) are described detailed in the application manual “Positioning”.

16.5 Conversions

The speeds can be converted into other speed formats using the formulas in this chapter:

Frequency [Hz] to	Speed [rpm]	See chapter 16.5.2
	Speed in user units [u/s]	See chapter 16.5.4
Speed [rpm] to	Frequency [Hz]	See chapter 16.5.1
	Speed in user units [u/s]	See chapter 16.5.6
Speed in user units [u/s]	Speed [rpm]	See chapter 16.5.5
	Frequency [Hz]	See chapter 16.5.3

16.5.1 Speed [rpm] to Frequency [Hz]

$$f\ [\mathrm{Hz}] = \frac{n[\mathrm{rpm}] \times No.of.Pole\ pairs\ (P.373)}{60}$$

16.5.2 Frequency [Hz] to Speed [rpm]

$$n[\mathrm{rpm}] = \frac{f\ [\mathrm{Hz}] \times 60}{No.of.Pole\ pairs\ (P.373)}$$

16.5.3 Speed in user units [u/s] to Frequency [Hz]

$$f\ [\mathrm{Hz}] = v\ [\tfrac{\mathrm{u}}{\mathrm{s}}] \times \frac{No.of\ pole\ pairs\ (P.373)}{Feed\ Constant\ (P.1115)} \times \frac{Gear\ Box: Motor\ Shaft\ Revolutions\ (P.1117)}{Gear\ Box: Driving\ Shaft\ Revolutions (P.1116)}$$

16.5.4 Frequency [Hz] to Speed in user units [u/s]

$$v\ [\tfrac{\mathrm{u}}{\mathrm{s}}] = f\ [\mathrm{Hz}] \times \frac{Feed\ Constant\ (P.1115)}{Polpaarzahl\ (P.373)} \times \frac{Gear\ Box: Driving\ Shaft\ Revolutions (P.1116)}{Gear\ Box: Motor\ Shaft\ Revolutions\ (P.1117)}$$

16.5.5 Speed in user units [u/s] to Speed [rpm]

$$n\ [rpm] = v\ [\tfrac{\mathrm{u}}{\mathrm{s}}] \times \frac{60}{Feed\ Constant\ (P.1115)} \times \frac{Gear\ Box: Motor\ Shaft\ Revolutions\ (P.1117)}{Gear\ Box: Driving\ Shaft\ Revolutions\ (P.1116)}$$

16.5.6 Speed [rpm] to Speed in user units [u/s]

$$v\ [\tfrac{\mathrm{u}}{\mathrm{s}}] = n\ [rpm] \times \frac{Feed\ constant\ (P.1115)}{60} \times \frac{Gear\ Box: Driving\ Shaft\ Revolutions\ (P.1116)}{Gear\ Box: Motor\ Shaft\ Revolutions\ (P.1117)}$$

16.6 Object support in the Software versions and EDS files

The support of CANopen was extended in various steps in the firmware. The following table lists, which objects are supported with the different software versions and the corresponding EDS file. Objects, that were added or where changes were made are marked in light blue color.

Long Object names are shortened sensible in the table to maintain the overview.

The EDS-files **BVACU52** and **BVACU52m** are identical in general.
In most cases the EDS file BVACU52.eds can be used.
BVACU52.eds reports device type "Frequency inverter" (Object 0x1000),
BVACU52m.eds reports device type "Servo" (Object 0x1000).
Since in particular cases "Servo" or "Frequency inverter" has to be reported from the EDS file to the PLC, both versions are available.
Both versions support the same objects.

Firmware	5.1.2+	5.1.5+	5.2.0+	5.3.0+	5.4.0+
EDS	BV_ACU .eds	BV_ACU2 .eds	BVACU52.eds BVACU52m.eds	BVACU53. eds	BVACU54. eds
0x1000 Device Type	x	x	x	x	x
0x1001 Error register	x	x	x	x	x
0x1005 COB-ID SYNC Object	x	x	x	x	x
0x1006 Comm. Cycle Period	x	x	x	x	x
0x1007 Syn. Window length	x	x	x	x	x
0x1008 Manuf. Device name	x	x	x	x	x
0x1009 Manuf. Hardw. Vers.	x	x	x	x	x
0x100A Manuf. Softw. Vers.	x	x	x	x	x
0x100C Guard Time	x	x	x	x	x
0x100D Life time factor	x	x	x	x	x
0x1010 Store parameters	x	x	x	x	x
0x1011 Restore parameters	x	x	x	x	x
0x1014 COB-ID emerg. object	x	x	x	x	x
0x1016 Consumer heartb. time	x	x	x	x	x
0x1017 Producer heartb. time	x	x	x	x	x
0x1018 Identity object	x	x	x	x	x
0x1029 Error behavior	x	x	x	x	x
0x1200 Server SDO param.	x	x	x	x	x
0x1400 RxPDO1 comm. param.	x	x 1)	x 1)	x 1)	x 1)
0x1401 RxPDO2 comm. param.	x	x 1)	x 1)	x 1)	x 1)
0x1402 RxPDO3 comm. param.	x	x 1)	x 1)	x 1)	x 1)
0x1600 RxPDO1 map. param.	x	x	x	x	x
0x1601 RxPDO2 map. param.	x	x	x	x	x
0x1602 RxPDO3 map. param.	x	x	x	x	x
0x1800 TxPDO1 comm. param.	x	x	x	x	x
0x1801 TxPDO2 comm. param.	x	x	x	x	x
0x1802 TxPDO3 comm. param.	x	x	x	x	x
0x1A00 TxPDO1 map. param.	x	x	x	x	x
0x1A01 TxPDO2 map. param.	x	x	x	x	x
0x1A02 TxPDO3 map. param.	x	x	x	x	x

1) Object 0x1400, 0x1401 and 0x1402 support since Firmware 5.1.5 Subindexes 3,4 and 5 like described in this manual. In previous Firmware Versions the described Functions like triggering a fault when exceeding the set “Event time” are not supported.

Firmware	5.1.2+	5.1.5+	5.2.0+	5.3.0+	5.4.0+
EDS	BV_ACU .eds	BV_ACU2 .eds	BVACU52.eds BVACU52m.eds	BVACU53. eds	BVACU54. eds
0x2nnn ACU parameter access	x	x	x	x	x
0x3000 Sync Jitter	x	x	x	x	x
0x3001 Digital In actual values	x	x	x	x	x
0x3002 Digital Out act. values	x	x	x	x	x
0x3003 Digital Out set values	x	x	x	x	x
0x3004 Boolean Mux	x	x	x	x	x
0x3005 Boolean DeMux	x	x	x	x	x
0x3006 Percentage Set value	x	x	x	x	x
0x3007 Percentage Act. value 1	x	x	x	x	x
0x3008 Percentage Act. value 2			x	x	x
0x3011 Act. Value Word 1			x	x	x
0x3012 Act. Value Word 2			x	x	x
0x3021 Act. Value Long 1			x	x	x
0x3022 Act. Value Long 2			x	x	x
0x3111 Ref. Value Word 1			x	x	x
0x3112 Ref. Value Word 2			x	x	x
0x3121 Ref. Value Long 1			x	x	x
0x3122 Ref. Value Long 2			x	x	x
0x5F10 Gear Factor				x	x
0x5F11 Phasing 1				x	x
0x5F12 Phasing 2				x	x
0x5F13 Phasing 3				x	x
0x5F14 Phasing 4				x	x
0x5F15 In Gear Threshold				x	x
0x5F16 In Gear Time				x	x
0x5F17 Position Controller				x	x
0x5F18 M/S Synchronization Offset					x
0x5FF0 Active motion block	x	x	x	x	x
0x5FF1 Motion block to resume	x	x	x	x	x

Firmware	5.1.2+	5.1.5+	5.2.0+	5.3.0+	5.4.0+
EDS	BV_ACU .eds	BV_ACU2 .eds	BVACU52.eds BVACU52m.eds	BVACU53. eds	BVACU54. eds
0x6007 Abort connect. option c.	x	x	x	x	x
0x603F Error code	x	x	x	x	x
0x6040 Controlword	x	x	x	x	x
0x6041 Statusword	x	x	x	x	x
0x6042 Target velocity	x	x	x	x	x
0x6043 Target velocity demand	x	x	x	x	x
0x6044 Control effort	x	x	x	x	x
0x6046 Velocity min max	x	x	x	x	x
0x6048 Velocity acceleration	x	x	x	x	x
0x6049 Velocity deceleration	x	x	x	x	x
0x604A Velocity quick stop	x	x	x	x	x
0x6060 Modes of Operation	x	x	x	x	x
0x6061 Modes of Op. display	x	x	x	x	x
0x6064 Position actual value	x	x	x	x	x
0x6065 Following error window	x	x	x	x	x
0x6066 Following error timeout	x	x	x	x	x
0x6067 Position Window	x	x	x	x	x
0x6068 Position Window time	x	x	x	x	x
0x606C Velocity act. value			x 2)	x 2)	x 2)
0x606D Velocity window			x 2)	x 2)	x 2)
0x606E Velocity window time			x 2)	x 2)	x 2)
0x606F Velocity Threshold			x 2)	x 2)	x 2)
0x6070 Velocity Threshold time			x 2)	x 2)	x 2)
0x6071 Target Torque	x	x	x	x	x
0x6077 Torque Actual value	x	x	x	x	x
0x6078 Current Actual value	x	x	x	x	x
0x6079 DC link circuit voltage	x	x	x	x	x
0x607A Target Position	x	x	x	x	x
0x607C Home Offset	x	x	x	x	x
0x6081 Profile Velocity	x	x	x	x	x
0x6083 Profile Acceleration	x	x	x	x	x
0x6084 Profile Deceleration	x	x	x	x	x
0x6085 Quick Stop deceleration	x	x	x	x	x
0x6086 Motion Profile type	x	x	x	x	x
0x6091 Gear ratio	x	x	x	x	x
0x6092 Feed constant	x	x	x	x	x
0x6098 Homing method	x	x	x	x	x
0x6099 Homing speeds	x	x	x	x	x
0x609A Homing acceleration	x	x	x	x	x
0x60C1 Interpol. Data record	x	x	x	x	x
0x60F4 Following err. Act. Val.	x	x	x	x	x
0x60F8 Max. Slippage			x 2)	x 2)	x 2)
0x60FF Target Velocity			x 2)	x 2)	x 2)

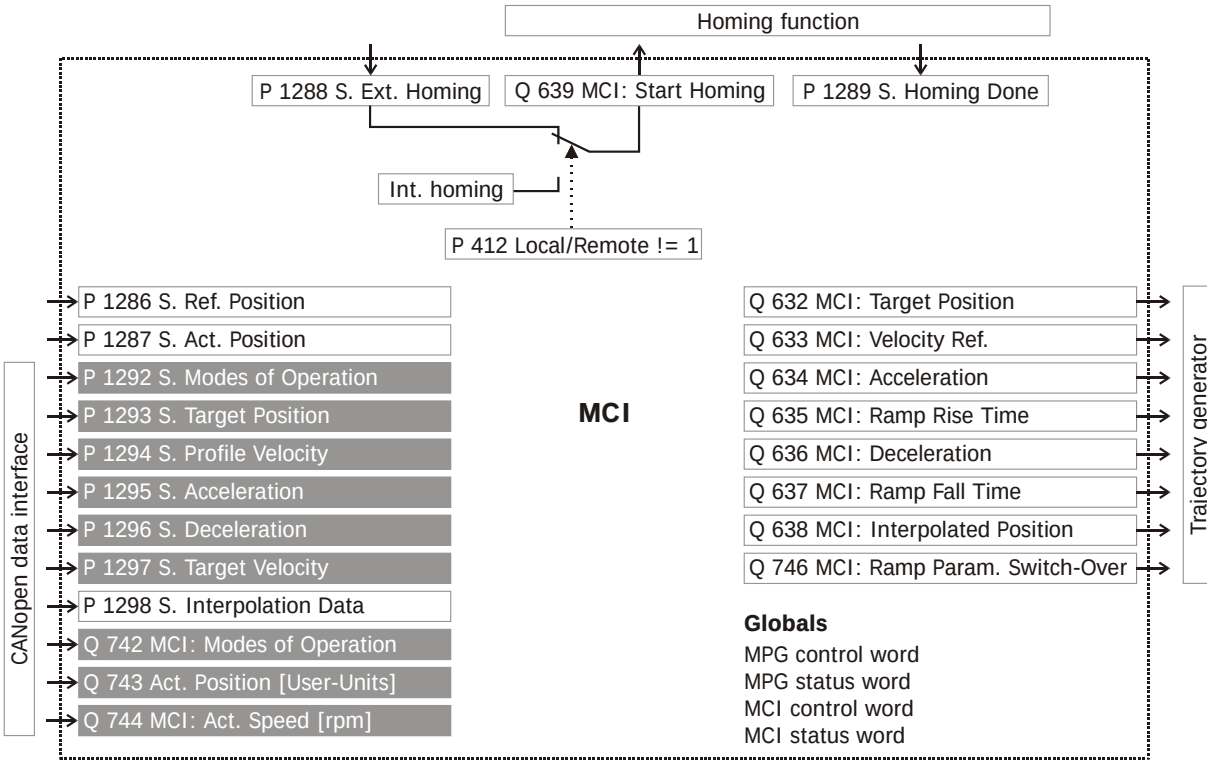
2) Profile Velocity Mode and the corresponding Objects were included in Version 5.2.0.

17 Motion-control-interface for Profibus connection

The motion control interface is the interface between the communication system and the motion control system. With factory setting the motion control interface is connected to the CANopen® system. This connection can be changed to the Profibus system.

As with CANopen® parameter *Local/Remote* **412** MUST be set to "1" = "control via state machine".
The motion control functions are available in configuration x**40** only.
This setting is necessary to control the inverter and motion control functions with the help of Control word (located in PZD1-OUT) and Status word (located in PZD1-IN). The functions and bit definitions are identical to the descriptions of CANopen®.

With Profibus the mode of operation "interpolated position mode" can **NOT** be used. In motion control configurations Profibus objects PZD2-OUT (reference speed) and PZD2-IN (actual speed) have no function.



MCI: motion control interface,
MPG: motion profile generator (trajectory generator),
P: parameter,
S: source,
rpm: revolution per minute

Only the grey marked parameters are necessary for controlling the motion control system by Profibus.

Input parameters **P xxxx** must be connected to Profibus OUT sources. These input parameters are direct accessible with parameter *Control Level* **28** set to **3**.

Output sources Q **xxx** must be connected to Profibus IN parameters (PZD-IN objects).

Index

Bonfiglioli worldwide network.

Bonfiglioli Australia
2, Cox Place Glendenning NSW 2761
Locked Bag 1000 Plumpton NSW 2761
Tel. (+ 61) 2 8811 8000 - Fax (+ 61) 2 9675 6605
www.bonfiglioli.com.au - sales@bonfiglioli.com.au

Bonfiglioli Brazil
Travessa Cláudio Armando 171 - Bloco 3
CEP 09861-730 - Bairro Assunção
São Bernardo do Campo - São Paulo
Tel. (+55) 11 4344 1900 - Fax (+55) 11 4344 1906
www.bonfigliolidobrasil.com.br
bonfigliolidobrasil@bonfiglioli.com

Bonfiglioli Canada
2-7941 Jane Street - Concord, Ontario L4K 4L6
Tel. (+1) 905 7384466 - Fax (+1) 905 7389833
www.bonfigliolicanada.com
sales@bonfigliolicanada.com

Bonfiglioli China
19D, No. 360 Pu Dong Nan Road
New Shanghai International Tower
200120 Shanghai
Tel. (+86) 21 5054 3357 - Fax (+86) 21 5970 2957
www.bonfiglioli.cn - bdssales@bonfiglioli.com.cn

Bonfiglioli Deutschland
Sperberweg 12 - 41468 Neuss
Tel. (+49) 02131 2988-0
Fax (+49) 02131 2988-100
www.bonfiglioli.de - info@bonfiglioli.de

Bonfiglioli España
TECNOTRANS BONFIGLIOLI S.A.
Pol. Ind. Zona Franca sector C, calle F, n°6
08040 Barcelona
Tel. (+34) 93 4478400 - Fax (+34) 93 3360402
www.tecnotrans.com - tecnotrans@tecnotrans.com

Bonfiglioli France
14 Rue Eugène Pottier BP 19
Zone Industrielle de Moimont II
95670 Marly la Ville
Tel. (+33) 1 34474510 - Fax (+33) 1 34688800
www.bonfiglioli.fr - btf@bonfiglioli.fr

Bonfiglioli India
PLOT AC7-AC11 Sidco Industrial Estate
Thirumudivakkam - Chennai 600 044
Tel. +91(0) 44 24781035 - 24781036 - 24781037
Fax +91(0) 44 24780091 - 24781904
www.bonfiglioliindia.com - bonfig@vsnl.com

Bonfiglioli Italia
Via Sandro Pertini lotto 7b
20080 Carpiano (Milano)
Tel. (+39) 02 985081 - Fax (+39) 02 985085817
www.bonfiglioli.it
customerservice.italia@bonfiglioli.it

Bonfiglioli New Zealand
88 Hastie Avenue, Mangere Bridge, Auckland
2022, New Zealand - PO Box 11795, Ellerslie
Tel. (+64) 09 634 6441 - Fax (+64) 09 634 6445
npollington@bonfiglioli.com.au

Bonfiglioli Österreich
Molkereistr 4 - A-2700 Wiener Neustadt
Tel. (+43) 02622 22400 - Fax (+43) 02622 22386
www.bonfiglioli.at
info@bonfiglioli.at

Bonfiglioli South East Asia
No 21 Woodlands indusrial park E1
#02-03 Singapore 757720
Tel. (+65) 6893 6346/7 - Fax (+65) 6893 6342
www.bonfiglioli.com.au
sales@bonfiglioli.com.sg

Bonfiglioli South Africa
55 Galaxy Avenue,
Linbro Business Park - Sandton
Tel. (+27) 11 608 2030 OR - Fax (+27) 11 608 2631
www.bonfiglioli.co.za
bonfigsales@bonfiglioli.co.za

Bonfiglioli Türkiye
Atatürk Organíze Sanayi Bölgesi,
10015 Sk. No: 17, Çigli - Izmir
Tel. +90 (0) 232 328 22 77 (pbx)
Fax +90 (0) 232 328 04 14
www.bonfiglioli.com.tr
info@bonfiglioli.com.tr

Bonfiglioli United Kingdom
Industrial Solutions
Unit 7, Colemeadow Road
North Moons Moat - Redditch,
Worcestershire B98 9PB
Tel. (+44) 1527 65022 - Fax (+44) 1527 61995
www.bonfiglioli.com
uksales@bonfiglioli.com
Mobile Solutions
3 - 7 Grosvenor Grange, Woolston
Warrington - Cheshire WA1 4SF
Tel. (+44) 1925 852667 - Fax (+44) 1925 852668
www.bonfiglioli.co.uk
mobilesales@bonfiglioli.co.uk

Bonfiglioli USA
3541 Hargrave Drive Hebron, Kentucky 41048
Tel. (+1) 859 334 3333 - Fax (+1) 859 334 8888
www.bonfiglioliusa.com

Bonfiglioli Vietnam
Lot C-9D-CN My Phuoc Industrial Park 3
Ben Cat - Binh Duong Province
Tel. (+84) 650 3577411 - Fax (+84) 650 3577422
www.bonfiglioli.vn
salesvn@bonfiglioli.com

Bonfiglioli

power, control and green solutions

Bonfiglioli has been designing and developing innovative and reliable power transmission and control solutions for industry, mobile machinery and renewable energy applicacations since 1956.

Bonfiglioli Riduttori S.p.A.
Via Giovanni XXIII, 7/A
40012 Lippo di Calderara di Reno
Bologna (Italy)

tel: +39 051 647 3111
fax: +39 051 647 3126
bonfiglioli@bonfiglioli.com
www.bonfiglioli.com

COD. VEC 567 R2