

ACTIVE CUBE

Communication Module

EtherNet/IP™

Frequency inverter 230 V / 400 V / 525 V / 690 V
0.25 kW ... 400 kW



TABLE OF CONTENTS

1	General Information about the Documentation	6
1.1	Instruction manuals	6
1.2	This document	7
1.3	Warranty and liability	7
1.4	Obligation	8
1.5	Copyright	8
1.6	Storage	8
1.7	Final decommissioning	8
2	General safety instructions and information on use	9
2.1	Terminology	9
2.2	Designated use	9
2.3	Misuse	10
2.4	Residual risks	10
2.5	Safety and warning signs on frequency inverter	10
2.6	Warning information and symbols used in the Operating Instructions	11
2.6.1	Hazard classes	11
2.6.2	Hazard symbols	11
2.6.3	Prohibition signs	11
2.6.4	Personal safety equipment	11
2.6.5	Recycling	11
2.6.6	Grounding symbol	12
2.6.7	ESD symbol	12
2.6.8	Information signs	12
2.6.9	Font style in documentation	12
2.7	Directives and guidelines to be adhered to by the operator	12
2.8	Operator's general plant documentation	12
2.9	Operator's/operating staff's responsibilities	12
2.9.1	Selection and qualification of staff	12
2.9.2	General work safety	12
2.9.3	Ear protectors	13
2.10	Organizational measures	13
2.10.1	General	13
2.10.2	Use in combination with third-party products	13
2.10.3	Handling and installation	13
2.10.4	Electrical connections	13
2.10.5	Safe operation	14
2.10.6	Maintenance and service/troubleshooting	14
2.10.7	Final decommissioning	15
2.11	Safety Instructions on Function "Safe Torque Off" (STO)	15
3	Storage and transport	17
3.1	Storage	17
3.2	Unpacking the device	17
4	Scope of supply	17
5	Introduction to module functionalities	18
6	Installation/Disassembly of the Communication Module	19
6.1	Installation	19

6.2	Disassembly.....	20
6.3	Connection of Unit.....	20
7	Commissioning	22
7.1	Switching on Mains Voltage	22
7.2	IP-Settings	23
7.3	TCP/IP Configuration Utility	23
7.3.1	VPlus.....	24
7.3.2	Module Info	24
7.4	OS Synchronization	25
7.4.1	Synchronization via CANopen	25
7.4.2	Synchronization via Systembus.....	26
8	Operational Behavior on Communication Failure	26
9	EtherNet/IP™ communication	27
9.1	How to create EDS Files	27
9.2	Supported configurations.....	29
9.3	Initialization time.....	30
9.4	EtherNet/IP™ overview	30
9.4.1	Class 0x01, Identity Object.....	30
9.4.2	Class 0xF5, TCP/IP Object	30
9.4.3	Class 0xF6, Ethernet Link Object	33
9.4.4	Class 0x04, Assembly Object.....	34
9.4.5	Class 0x0F, Parameter Object	34
9.4.6	Class 0x28, Motor Data Object.....	42
9.4.7	Class 0x29, Control Supervisor Object.....	42
9.4.8	Class 0x2A, AC/DC-Drive Object	43
9.5	Operating behavior in the case of bus connection failure.....	43
9.6	Access to Inverter Parameters (Classes 0x64 ... 0x73)	44
9.7	I/O Assemblies.....	45
9.7.1	CIP Assemblies 20/21/22/24 + 70/71/72/74.....	46
9.7.2	Vendor Specific Assemblies 100/101.....	51
9.7.3	Handling of I/O Assemblies, Setup Device/PLC	54
9.8	Sending Explicit Messages from the PLC Program.....	55
9.8.1	How to create a MESSAGE Tag	55
9.8.2	PLC Program Code Examples	58
9.9	Diagnosis of EtherNet/IP™ state	60
10	CANopen Overview.....	61
10.1.1	Communication Objects	61
10.1.2	Application Objects	61
10.1.3	SDO Function	61
10.1.4	PDO Function.....	65
10.1.5	Emergency Function	65
10.1.6	SYNC (synchronous transmission)	66
10.1.7	NMT Functions.....	66
10.1.8	Guarding	68
10.1.9	Heartbeat.....	69
10.2	Special CANopen objects.....	69
10.2.1	0x1005/0 COB-ID SYNC Message	70
10.2.2	0x1006/0 Communication Cycle Period.....	70

10.2.3	0x1007/0 Synchronous window length	71
10.2.4	0x100C/0 Guard Time	71
10.2.5	0x100D/0 Lifetime Factor	71
10.2.6	0x1014/0 COB-ID Emergency Message	71
10.2.7	0x1016/n Consumer Heartbeat Time.....	72
10.2.8	0x1017/0 Producer Heartbeat Time	72
10.2.9	0x1029/n Error Behavior	73
10.2.10	0x1200/n SDO Server Parameter	73
10.2.11	0x1400/n, 0x1401/n, 0x1402/n RxPDO Communication Parameters	73
10.2.12	0x1800/n, 0x1801/n, 0x1802/n TxPDO Communication Parameters	74
10.2.13	0x3000/0 SYNC Jitter.....	75
10.3	CANopen objects	76
10.3.1	Objects tabular overview	76
10.3.2	Communication Objects (0x1nnn)	84
10.3.3	Manufacturer objects (0x2nnn) – Parameter access	88
10.3.4	Manufacturer objects (0x3000 ... 0x5FF1)	90
10.3.5	Device Profile Objects (0x6nnn)	105

1 General Information about the Documentation

1.1 Instruction manuals

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

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 frequency inverter by the software.

Operating Instructions

The Operating Instructions document 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.

Application manual

The application manual supplements the documentation for purposeful installation and commissioning of the frequency inverter. Information on various topics in connection with the use of the frequency inverter is described in context with the specific application.



If you need a copy of the documentation or additional information, contact your local representative of BONFIGLIOLI.

The following instructions are available for the *ACTIVE CUBE* series:

<i>ACTIVE CUBE</i> Operating Instructions	Function of frequency inverter.
Quick Start Guide <i>ACTIVE CUBE</i>	Installation and commissioning Supplied with the device.
Manuals Communication interfaces	CM-CAN: CANopen manual CM-PDP-V1: Profibus DP-V1 manual CM-232/CM-485: VABus manual (serial protocol) CM-232/CM-485 Modbus: Modus ASCII and RTU manual CM-VABus/TCP: Ethernet Module CM-VABus/TCP CM-ModbusTCP: Ethernet Module CM-Modbus/TCP CM-EtherCAT®: Ethernet Module CM-EtherCAT® CM-ProfiNet: Ethernet Module CM-ProfiNet CM-Ethernet/IP: Ethernet Module CM-Ethernet/IP
Manuals Extension modules	EM-ABS-01: Absolute encoder module EM-ENC-01: Speed sensor (encoder) module EM-ENC-02: Speed sensor (encoder) module EM-ENC-03: Speed sensor (encoder) module EM-ENC-04: Speed sensor (encoder) module EM-ENC-05: Speed sensor (encoder) module EM-IO-01: Extension module for digital inputs/outputs EM-IO-02: Extension module for digital inputs/outputs EM-IO-03: Extension module for digital inputs/outputs EM-IO-04: Extension module for digital inputs/outputs EM-RES-01: Resolver module EM-RES-02: Resolver module EM-RES-03: Resolver module EM-SYS: System Bus module
Safe Torque Off (STO) manual	Safety function STO
Liquid Cooling - Complement to Operating Instructions	Properties specific to liquid cooled frequency inverters
Application manual "Parallel connection"	Parallel connection of Size 8 frequency inverters
PLC application manual	Logic linking of digital signals. Functions for analog signals such as comparisons and mathematical functions. Graphical support for programming with function blocks.
Application manual "Positioning"	Positioning functions of Configurations x40.
Application manual "Electronic gear"	Linking of at least 2 drives as electronic gear with Slave drive in Configuration x15 or x16.
Application manual "Hoist unit drives"	Advanced brake control for hoist unit drives.

The present documentation was prepared with great care and it was subjected to extensive and repeated reviews. For reasons of clarity, it was not possible to include all details of all types of the product in the documentation. Neither was it possible to consider all conceivable installation, operation or maintenance situations. If you require further information or if you meet with specific problems which are not dealt with in sufficient detail in the documentation, contact your local BONFIGLIOLI agent. The present document was created in German. Other language versions are translations.

1.2 This document

The present user manual of the CM-Ethernet/IP communication module complements the Operating Instructions and the "Quick Start Guide" for the frequency inverters of the ACU device series.

The user manual contains important information on the installation and use of the EtherNet/IP™ communication module CM-Ethernet/IP 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 life of the frequency inverter.

For this reason, make sure you read the user manual carefully.

IMPORTANT:

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.



For safe commissioning and operation of the ACU (ACTIVE Cube) series, the following documentation must be complied with:

- The Operating Instructions Document
- Application manual "Safe Torque Off ACU"

This document applies to the following frequency inverter series:

- ACTIVE Cube 210
- ACTIVE Cube 410
- ACTIVE Cube 510
- ACTIVE Cube 610

1.3 Warranty and liability

BONFIGLIOLI Vectron GmbH (hereinafter referred to as "manufacturer") notes that the contents of this Operating Instructions document do not form part of any previous or existing agreement, assurance or legal relationship between the manufacturer and the user of these Operating Instructions (hereinafter referred to as the "User"). 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 prior notice. The manufacturer assumes no responsibility to update these Operating Instructions. The manufacturer shall not be liable for any damage, injuries or costs which may be caused by the aforementioned reasons.

In addition, the manufacturer excludes any warranty and disclaims all liability, including without limitation direct, indirect, special, punitive, incidental, exemplary or consequential damages arising out of or in connection with 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 frequency 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.

1.4 Obligation

These Operating Instructions 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 Operating Instructions and, in particular, the safety instructions in order to prevent personal and material losses.

1.5 Copyright

In accordance with applicable law any copyrights relating to this document shall remain with
BONFIGLIOLI Vectron GmbH
Europark Fichtenhain B6
47807 Krefeld
Germany

This document is intended for the operator of the frequency inverter. Any disclosure or copying of this document, exploitation and communication of its contents (as hardcopy or electronically) shall be forbidden, unless permitted expressly.

Any non-compliance will constitute an offense against the copyright law, the law against unfair competition and the German Civil Code and may result in claims for damages. All rights relating to patent, utility model or design registration reserved.

1.6 Storage

The documentation forms an integral part of the frequency inverter. It must be stored such that it is accessible to operating staff at all times. If the frequency inverter is sold on to other users, then the documentation must also be handed over.

1.7 Final decommissioning

After the end of product service life, the user/operator must take the device out of operation.



For more information about the decommissioning of the device refer to the applicable operating instructions document.

Disposal requirements under European Union WEEE regulations

The product is marked with the WEEE symbol shown below.

This product cannot be disposed as general household waste. Users responsible for the final disposal must make sure that it is carried out in accordance with the European Directive 2012/19/EU, where required, as well as the relative national transposition rules. Fulfil disposal also in according with any other legislation in force in the country.



2 General safety instructions and information on use

This chapter 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.

Skilled Personnel

The term Skilled Personnel covers staff that are 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, Skilled Personnel 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 2014/35/EU and DIN EN 61800-5-1. CE-labelling is based on these standards. Responsibility for compliance with the EMC Directive 2014/30/EU 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,
- when general requirements, such as operating conditions and technical data, are not met.

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

Explosion protection

The frequency inverter is an IP 20 ingress protection rating 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 a 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 in frequency inverter if no external disconnection device was installed by the operator.

During operation, all covers must be installed correctly, and all electrical cabinet doors must be closed to minimize electrical hazards.

When LEDs and other indicating elements on the frequency inverter go out, this does not necessarily mean that the device is deenergized. Before carrying out any Work at the device where contact with energized parts might be possible, it must be checked in any case, i.e. irrespective of the status of any indicating elements that may be installed, if the device is deenergized.

Charged capacitors in DC link

Sizes 1 through 7 (up to 160 kW): The DC-link may have dangerous voltage levels even up to 3 minutes after shutdown.

Size 7 and 8 (as from 160 kW): The DC-link may have dangerous voltage levels even up to 10 minutes after shutdown.

Electrostatic charging

Touching electronic components entails the risk of electrostatic discharges.

Thermal hazards

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

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

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

2.5 Safety and warning signs on 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 Operating Instructions

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.

NOTICE

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
	Danger of crushing		

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
	Wear ear protectors

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 Sensitive Devices, i.e. components and assemblies sensitive to electrostatic energy

2.6.8 Information signs

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

2.6.9 Font style in documentation

Example	Font style	Use
1234	bold	Representation of parameter numbers
<i>Parameter</i>	inclined, font: Times New Roman	Representation of parameter names
P.1234	bold	Representation of parameter numbers without name, e.g. in formulas
Q.1234	bold	Representation of source numbers

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.
- For liquid cooled frequency inverters, comply with the cooling water guideline VGB-R 455 P.
- 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 Operating Instructions, the operator should issue separate internal user manuals for the frequency inverter. The Operating Instructions of the frequency inverter must be included in the Operating Instructions 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 skilled personnel. The staff must not be under the influence of any drugs. Note the minimum age required by law. Define the staff's responsibility pertaining to 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 Operating Instructions 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 Operating Instructions, 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.

2.9.3 Ear protectors

- The frequency inverter produces noise. Due to noise development, frequency inverters should only be installed in normally unstaffed areas.
- Noise emission in operation is < 85 dB(A) in the case of sizes 1 through 7.
- Noise emission in operation is approx. 86 dB(A) in the case of size 8. Ear protectors must be used when staying near the frequency inverter.

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 offers 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 so at your own risk.

2.10.3 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 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.

2.10.4 Electrical connections

- The five safety rules must be complied with.
- Never touch live terminals. In sizes 1 through 7, the DC-link may have dangerous voltage levels up to 3 minutes after shutdown. In size 8, the DC-link may have dangerous voltage levels up to 10 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 in which 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. The frequency inverter may be operated in TN, TT and IT grid types. Precautions must be taken for operation in IT grids, see Chapter 7 "Electrical installation". Operation in a corner-grounded TN grid shall not be permissible.

2.10.4.1 The five safety rules

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

- Disconnect
- Secure to prevent restarting
- check for absence of voltage,
- carry out earthing and short-circuiting
- cover or shield neighboring live parts

2.10.5 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, all covers must be installed correctly, and all electrical cabinet doors must be closed. During operation, never open the machine/plant.
- No connection work shall be carried out while power supply is on.
- 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 braking 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. After shutdown, wait for at least 3 minutes (sizes 1 through 7) and at least 10 minutes (size 8) before starting any 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 skilled personnel 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 the frequency inverter and children must not have access to the device.
- Do not bypass nor decommission any protective devices.
- 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 AutoStart function is activated.
If staff are 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).

2.10.6 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. Any repair work may only be carried out by the manufacturer or persons approved/licensed by the manufacturer. Any repair work must be carried out by qualified electricians. 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.

2.10.7 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.

2.11 Safety Instructions on Function "Safe Torque Off" (STO)

The function „Safe Torque Off" (STO) is a functional safety feature, i.e. it protects staff from damage, provided that projecting, installation and operation are performed properly. This function does not disconnect the plant from power supply.

In order to disconnect the plant from power supply (e.g. for maintenance work), an "Emergency Stop" provision as per EN 60204 must be installed.

WARNING



Uncontrolled Starting

Improper installation of the safety circuitry may result in uncontrolled starting of the drive. This may cause death, serious injuries and significant material damage.

- Safety functions may only be installed and commissioned by skilled personnel. The STO function is not suitable for emergency stop as per EN 60204. An emergency stop can be realized by installing a mains contactor.

An emergency stop according to EN 60204 must be functioning in all operation modes of the frequency inverter. Resetting of an emergency stop must not result in uncontrolled starting of the drive.

The drive is started again when the function STO is no longer required. In order to comply with EN 60204, it must be ensured by taking external measures that the drive does not start without prior confirmation.

Without a mechanical brake, the drive will not stop immediately but coast to a standstill. If this may result in personal or material damage, additional safety measures must be taken.

- If persons may be endangered after disconnection of the motor power supply by STO, access to the hazard areas must be prevented until the drive has stopped.
- Check the safety function at regular intervals according to the results of your risk analysis. Bonfiglioli Vectron GmbH recommends that the check be performed after one year, at the latest.

The STO function is fail-safe for one fault. However, on rare occasions, the occurrence of component defects may cause jerking of the motor shaft (max. 180°/pole pair, e. g. jerk by 90° with 4-pole motor, 180°/2).

- Check if this causes a dangerous movement of the machine.
- If the STO function is used, the special safety, installation and instructions on use instructions shall be complied with.

WARNING**Dangerous voltage!**

The safety function "Safe Torque Off" may only be used if mechanical work is to be performed on the driven machines, not for work on live components.

After disconnection of an external DC 24 V power supply, the DC link of the frequency inverter is still connected to mains supply.

Even if power supply to the motor is disconnected, and the motor is coasting to a standstill or has already stopped, high voltages may still be present on the motor terminals.

Before working (e. g. maintenance) on live parts, the plant must always be disconnected from mains supply (main switch). This must be documented on the plant.

When the function "Safe Torque Off" is triggered, the motor is not isolated from the DC link of the frequency inverter. High voltage levels may be present at the motor.

- Do not touch live terminals.



The application manual "Safe Torque Off STO" must be complied with, particularly if the safety function described there is used.

3 Storage and transport

3.1 Storage

NOTICE

Damage caused by incorrect storage

- Wrong or inappropriate storage may result in damage, e.g. due to moisture and dirt. Avoid major temperature variations and high air humidity.
 - During storage, protect the device against moisture and dirt.
-
- The frequency inverters must be stored in an appropriate way. During 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 small 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. After one year of storage, connect the device to mains voltage for 60 minutes.

3.2 Unpacking the device

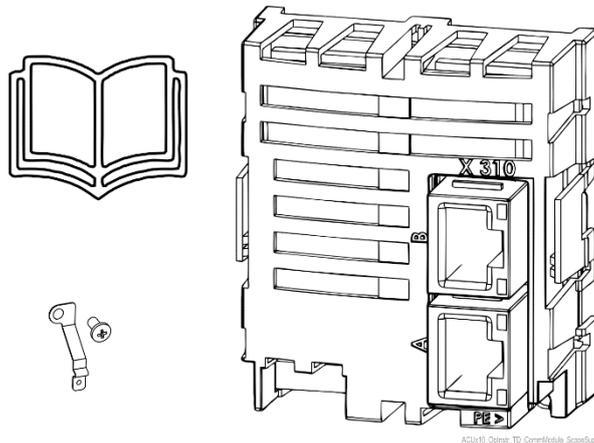
- Carefully remove packaging.
- Check if the delivered devices corresponds to the order.
- Check the device for transport damage and completeness.
- Any defects/damage must be reported to the supplier immediately.



Ensure that all packaging materials are disposed of in an environmentally compatible manner.

4 Scope of supply

The scope of delivery described can be supplemented by optional components and adapted to the customer-specific requirements.



5 Introduction to module functionalities



Software or Hardware designations and label names of the companies mentioned are usually protected by law concerning trademark, patent and brand name. Licensing right and copyright are held by the respective companies. Contents of this document may not be copied or duplicated outside the legal regulations without written permission.

The Ethernet-IP implementation is 100% compatible with the Common Industrial Protocol (CIP) specification.

Ethernet-IP communication requires software version 5.4.0 (ACUx01), 7.0.1.0 (ACUx10) or higher.

Ethernet-IP communication is available with module:

- Communication module CM-Ethernet/IP

The CM-Ethernet/IP module is enclosed with the frequency inverter as a separate component and must be fitted by the user. This is described in detail in Chapter 6 "Installation/Disassembly of the Communication Module".



These instructions are not to be understood as fundamental information on EtherNet/IP™. They presuppose underlying knowledge of the methods and mode of effect of EtherNet/IP™ 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.

This document shows how to connect the hardware, shows the relevant parameters and the available objects.

6 Installation/Disassembly of the Communication Module

6.1 Installation

The communication module CM-Ethernet/IP is a ready-to-use optional module. Additionally, a PE spring is enclosed for PE connection (shield).

CAUTION



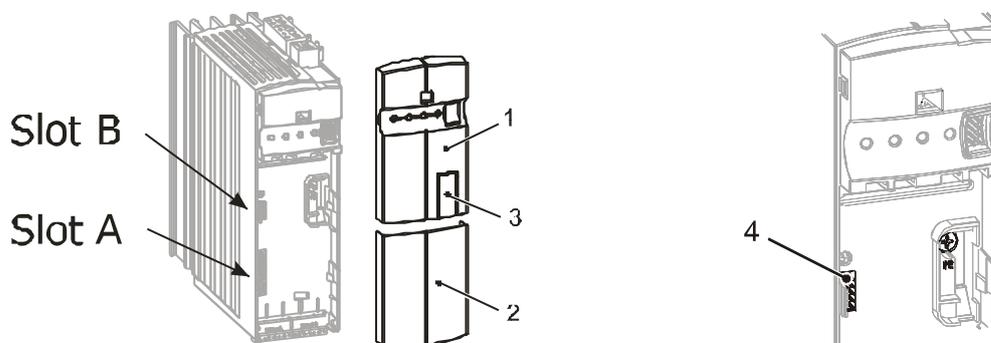
Component damage

The control terminals may be energized.

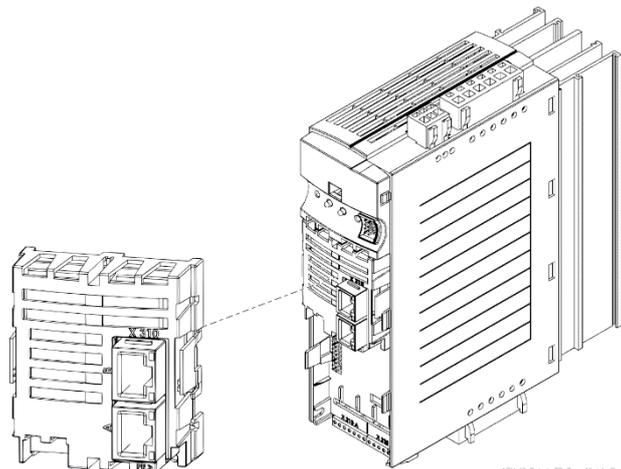
- The unit may only be connected with the power supply switched off.
- Verify safe isolation from power supply.
- Switch off power supply before connecting or disconnecting the control inputs and outputs. Otherwise, components may be damaged.

Work steps:

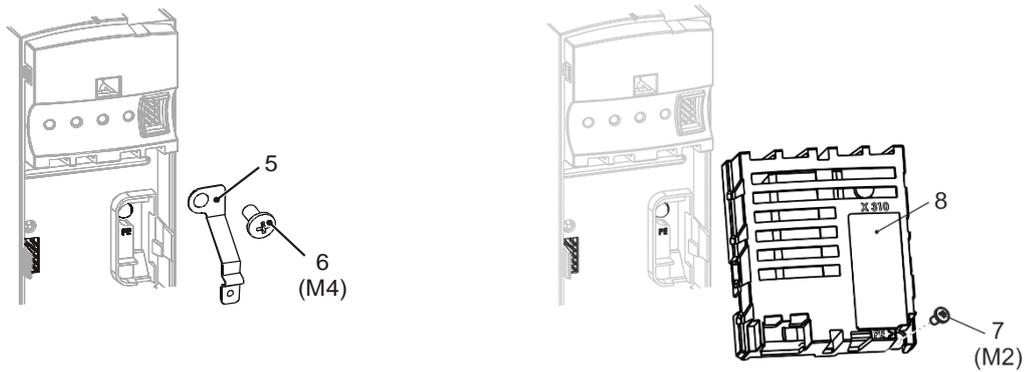
- Disconnect the frequency inverter from the mains voltage and protect it against being energized unintentionally.
- Remove covers **(1)** and **(2)** of the frequency inverter. Slot B **(4)** for the communication module is now accessible.



- Mount the supplied PE spring **(5)** using the M4 screw **(6)** which is mounted on the inverter. The spring must be aligned centrally.
- Insert the communication module in slot B **(4)** until it engages audibly.



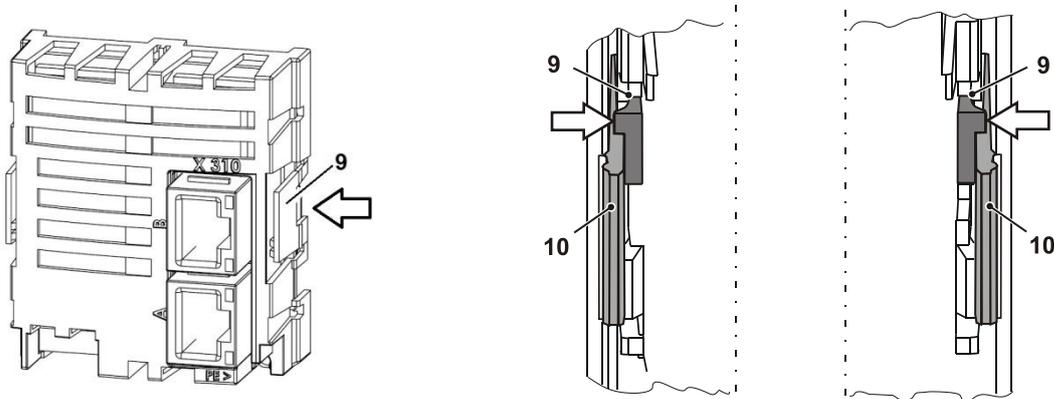
- Fix the communication module by screwing the M2 screw **(7)** of the module to the PE spring **(5)**.



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

6.2 Disassembly

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



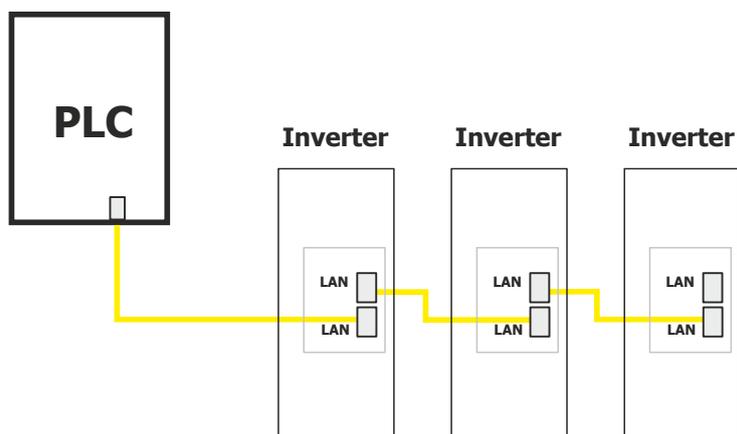
- Loosen the M2 screw **(7)** on the communication module
- Unplug the communication module from Slot B **(4)** by unlocking the locking hooks **(9)** on the right and left hand side of the module from the case of the frequency inverter using a small screwdriver.

The locking hooks **(9)** are located at the place where the locking 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 (\leftarrow). 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 (\Rightarrow).
- 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)**.

6.3 Connection of Unit

The EtherNet/IP™ module is connected to PLC or switch using RJ45 connectors (LAN) with standard twisted-pair Ethernet cables.



ANG410-TD-EM-AUT1-V01

Figure 6-1: EtherNet/IP™ topology

The EtherNet/IP™ module supports switch functionality for easy daisy-chaining of inverters.



The transmission speed of the EtherNet/IP™ communication module EtherNet/IP™ is 10/100 MB and the maximum cable length between two nodes is 100 m.

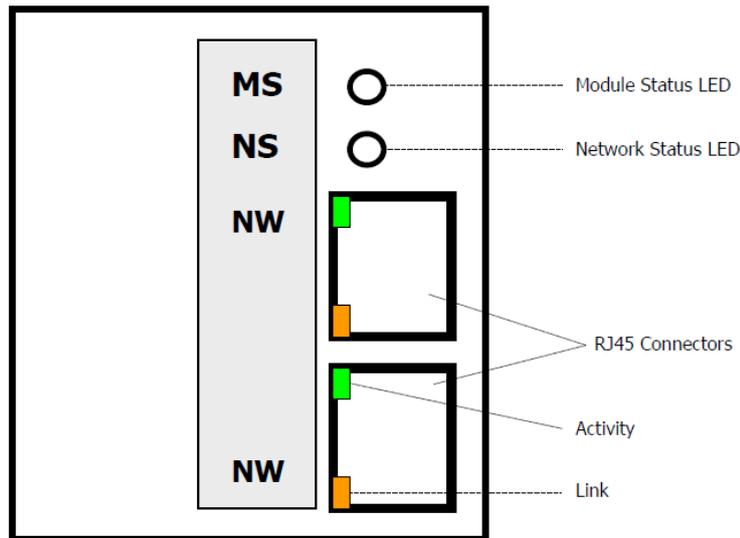
7 Commissioning

7.1 Switching on Mains Voltage

After completion of the installation work, make sure to check all control and power connections again before switching on the mains voltage. When all electrical connections are correct, make sure that the frequency inverter is not enabled (control inputs S1IND/STOA and S7IND/STOB open). After power-up, the frequency inverter carries out a self-test and the relay output (X10) reports "Fault".

After a few seconds, the self-test is complete, the relay (X10) picks up and signals "no fault".

If the unit is in "as-delivered" condition or after resetting the unit to the factory settings, the guided commissioning procedure is started automatically. On the control unit, the "SetUP" menu from the menu branch CTRL is displayed.



MS = Module Status, NS = Network Status, NW = Network

Figure 7-1: Ethernet connector

LED status indicators

The module status LED (MS) indicates the current status of the module.

LED Status	Module status
Off	Module is off.
On	Module is on and running.

The following states are defined:

Indicator State	Meaning	Description
Steady Off	No power	No power is supplied to the interface.
Steady Green	Device operational	The interface is operating correctly.
Flashing Green	Standby	The interface has not been configured.
Flashing Red	Minor fault	The interface has detected a recoverable minor fault. NOTE: An incorrect or inconsistent configuration would be considered a minor fault. Remedy: power off and on the module.
Steady Red	Major fault	The module has detected a non-recoverable major fault.
Flashing Green / Red	Self-test	The module is performing its power-up testing.

The Network Status LED (NS) indicates the current status of the connection.

LED Status	Module status
Off	Not powered
Flashing	No Ethernet connection or no cyclic exchange of data is taking place.
On	Exchange of Cyclic data ongoing.

The following states are defined:

Indicator State	Meaning	Description
Steady Off	Not powered, no IP address	The device is powered off, or is powered on but with no IP address configured (Interface Configuration attribute of the TCP/IP Interface Object).
Flashing Green	No connections	An IP address is configured, but no CIP connections are established, and an Exclusive Owner connection has not timed out.
Steady Green	Connected	At least one CIP connection (any transport class) is established, and an Exclusive Owner connection (defined Volume 1: Ethernet/IP Adaption of CIP, Chapter 2) has not timed out.
Flashing Red	Minor fault	An Exclusive Owner connection (defined Volume 1: Ethernet/IP Adaption of CIP, Chapter 2) for which this device is the target has timed out. The network status indicator shall return to steady green only when all timed out Exclusive Owner connections are reestablished.
Steady Red	Major fault	For devices that support duplicate IP address detection, the device has detected that its IP address is already in use.
Flashing Green / Red	Self-test	The module is performing its power-up testing.

7.2 IP-Settings

The IP-Address and other network parameters can be set in the Communication – Ethernet branch of the VPlus Parameter window or with the VPlus TCP/IP Configuration Utility:

Parameter		Setting		
No.	Description	Min.	Max.	Fact. sett.
1432	IP-Address	0. 0. 0. 0	255. 255. 255. 255	172. 22. 1. 25
1433	Netmask	0. 0. 0. 0	" " " "	255.255.255. 0
1434	Gateway	0. 0. 0. 0	" " " "	0. 0. 0. 0
1435	DNS Server	0. 0. 0. 0	" " " "	0. 0. 0. 0
1436	DHCP Option	0 - Disabled	1 - Enabled	0 - Disabled

7.3 TCP/IP Configuration Utility

- Ensure your control device (PC) is connected to the Ethernet network.
- Start the TCP/IP Configuration Utility.

The configuration dialogue is displayed as follows:

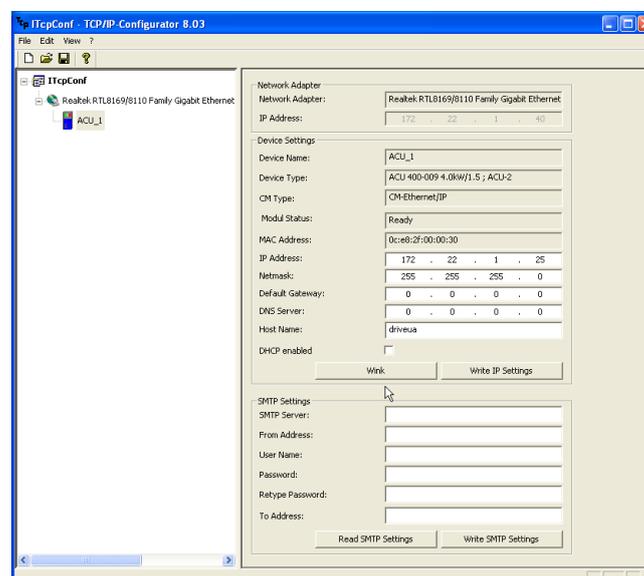


Figure 7-2: TCP/IP Configuration

- Select the Ethernet interface as required.
- Select the inverter as required.
- Enter the network settings according to your required configuration.
- Apply the settings with the *Write IP Settings* command button.

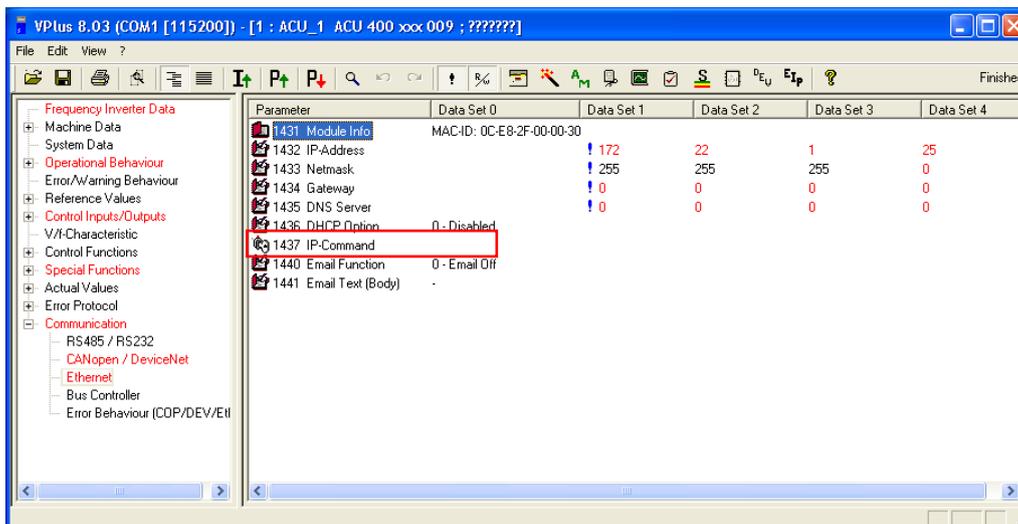
Wink command

The Wink Command can be sent from the TCP/IP Configuration Utility by clicking the "Wink" button. This will cause the Inverter's LEDs (green and red) to blink for 10 seconds.

7.3.1 VPlus

- Ensure your control device (PC) is connected to the Ethernet network.
- Start the VPlus software Utility.
- Select the Ethernet interface as required.
- Select the inverter as required.

The parameter setting dialogue is displayed as follows:



- Search for the parameters listed in the table above or select the branch *Communication* → *Ethernet* in the parameter window of the VPlus dialogue.
- Enter the network settings according to your required configuration.

NOTE

Risk of component damage!

If a maximum number of write cycles for the EEPROM is exceeded, it will be destroyed. The values are entered automatically in the EEPROM of the controller. However, only a limited number of write cycles is permissible for the EEPROM (approx. 1 million cycles).

- If cyclic writing of data is required, use the RAM.

- Apply the chosen settings via parameter *IP-Command* **1437** = 0 (Apply). This command sends the settings to the EtherNet/IP™ Module and writes them to the inverter's EEPROM.

If settings have been changed but **not applied**, you can reload the last settings saved via parameter *IP-Command* **1437** = 1 (Reload).

7.3.2 Module Info

EM-Software version **016** shows the Software version of the CM module.

Parameter *Module Info* **1431** shows basic Ethernet based module data:

MAC-ID:	unambiguous MAC-ID
Sno:	Serial number
HW:	Hardware version
SW:	Ethernet Software version

7.4 OS Synchronization

The operating system (OS) of the frequency inverter can be synchronized with a PLC or other device. Synchronization of the operating system will improve the operating characteristics of the machine. Synchronization is used to eliminate CPU **phase** shifting between master and slave devices to make sure that calculations are carried out at the same time.

Note, that only small deviations of the CPU clock frequencies between devices (i.e. different CPU Quartz clock frequencies) of $\pm 1 \text{ ‰}$ can be compensated.

Synchronization via CANopen®:

If CANopen is used, synchronization to a CANopen PLC can be turned on or off. Synchronization is done via CANopen SYNC telegrams.

Synchronization via systembus:

If CANopen is used simultaneously with systembus, synchronization can be done either on CANopen, systembus or turned off. Synchronization is effected through systembus SYNC telegrams or systembus RxPDO telegrams.



If the operating system is synchronized via CANopen, the CANopen master must support the CANopen synchronization mechanisms.



Synchronization via EtherNet/IP™ is not possible.

OS_SyncSource 1452

Operation mode	Function
0 - Auto	The synchronization source is selected automatically by the frequency inverter. Factory setting.
1 - CANopen	The OS is synchronized via CANopen®.
2 - Systembus	The OS is synchronized via Systembus.
3 - Ind. Ethernet Module	The OS is synchronized via Ethernet Module
4 - Synchronised Ind. Ethernet Module	The OS is synchronized via synchronized Ethernet Module e.g. EthernetIP
99 - Off	The OS is not synchronized with other devices.

Auto mode: Selection is done based on the decision table:

Fieldbus active	Systembus active	Synchronization
Yes	Yes	Synchronisation via Fieldbus
Yes	No	
No	Yes	Synchronization via Systembus
No	No	No Synchronization activated.

1453 *OS SyncSource Act* shows the active Synchronization source.

7.4.1 Synchronization via CANopen

Status "Synchronization via CANopen active" is identified via parameter setting **387** *CAN Node Number* > 1 and a running synchronous PDO.

The parameter **1451** *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 behavior.

Parameter		Setting		
No.	Description	Min.	Max.	Factory setting
1451	OS Synctime	700 us	900 us	800 us

7.4.2 Synchronization via Systembus

Status "Synchronization via systembus active" is identified via parameter setting **900** *Node ID* > 1. In addition, parameter **1180** *Synchronization* must be set to SYNC or RxPDO.

The source of the operating system (OS) synchronization is set via **1180** *Operation mode*. This defines the Sync event (RxPDO or SYNC telegram), which will be used for synchronization of PDOs:

930 *TxPDO1 Function*
932 *TxPDO2 Function*
934 *TxPDO3 Function*

936 *RxPDO1 Function*
937 *RxPDO2 Function*
938 *RxPDO3 Function*

Synchronization Operation mode 1180		
Operation mode		Function
0 -	Off	Synchronization via systembus is deactivated. Factory setting.
1 -	RxPDO1	Synchronization via systembus is activated via RxPDO1.
2 -	RxPDO2	Synchronization via systembus is activated via RxPDO2.
3 -	RxPDO3	Synchronization via systembus is activated via RxPDO3.
10 -	SYNC	Synchronization via systembus is activated via SYNC.

8 Operational Behavior on Communication Failure

The operational behavior of the EtherNet/IP™ system failures due to communication failure or expected packet rate can be parameterized. The required behavior is set with parameter *Error Behavior* **388**.

The warning message is displayed by the LED's and can be read out on the operating unit via parameter *Warnings* **269** or transmitted via one of the digital control outputs.

<i>Error Behavior</i> 388	Function
0 -No Reaction	Operating point is maintained
1 -Error	"Fault" status will be activated immediately. Factory setting.
2 -Stop	Control command "Disable voltage" and switch to "switch on disabled" status.
3 -Quick-Stop	Control command "Quick stop" and switch to "switch on disabled" status.
4 -Ramp-Stop + Error	Control command "Disable operation" and switch to "Error" status once the drive has been shut down.
5 -Quick-Stop + Error	Control command "Quick stop" and switch to "Error" status once the drive has been shut down.

NOTE

The parameter settings *Bus Error Behaviour* **388** = 2 ... 5 are evaluated depending of parameter *Local/Remote* **412**. For details see chapter 9.2 and 10.3.5.1.

There are numerous options of parameterizing the fault and warning behavior of the frequency inverter. For details about possible faults, refer to the co-applicable operating instructions document.

9 EtherNet/IP™ communication



The EtherNet/IP™ implementation in Bonfiglioli inverters is compatible with the ODVA/CIP EtherNet/IP™ specification.

Common Industrial Protocol (CIP™) makes a distinction between input and output assemblies. "input" and "output" in this context are viewed **from the perspective of the controlling element** (e.g., a PLC/PAC). An input assembly in a device collects data from the input application (e.g., field wiring terminal, proximity sensor, etc.) and produces it on the network, where it is **consumed** by the controlling device and/or operator interface. An output assembly in a device consumes data that the controlling element sends to the network and writes that data to the output application (e.g., field wiring terminals, motor speed control, etc.).

The following chapters describe the implemented classes/instances/attributes of EtherNet/IP™. For I/O data exchange, according to the CIP-defined AC drives device profile, output assemblies 20/21/22/24 and input assemblies 70/71/72/74 are available.

In addition, the vendor-specific output assembly 100 and input assembly 101 are available. These assemblies support objects defined in the Bonfiglioli CANopen implementation, including "dynamic mapping" as known from CANopen Rx/TxPDO mapping.



All of the CANopen functions are available, **with the exception of** motion control "*interpolated position mode*" and "*cyclic synchronous position mode*".

EtherNet/IP™ supports two types of connection:

- Explicit Messaging - confirmed services for get/set values
- I/O - confirmed/unconfirmed services for R/W process data

Explicit Messaging

All values of classes/instances/attributes are accessible by explicit message (get/read or set/write, from the PLC's point of view).

I/O

For I/O, different types of communication are available:

- Polled
- Change of state
- Cyclic

Polled

With the "Polled" I/O setting, the PLC sends data OUT to the inverter. After receiving OUT data the inverter sends IN data to the PLC.

Change of state

With the "Change of state" (COS) setting, the PLC sends OUT data whenever the OUT data value has changed. The inverter sends IN data whenever the IN data value has changed. To minimize sending repeated messages in short time refer to Chapter 9.4.5.9 "Change of State (COS) Mask, Parameters 34 to 39".

Cyclic

With the "Cyclic" setting, the PLC sends OUT data cyclically to the inverter. The inverter sends IN data cyclically to the PLC.

I/O-Connections supported in ANG:

- PLC - output data - Polled
- Inverter - input data - Polled, Change-of-State, Cyclic

9.1 How to create EDS Files

EtherNet/IP™ uses type-specific EDS files for each type of inverter (size, voltage, current). These EDS files are required for configuration utilities, e.g. Rockwell RSLogix 5000.

EDS files are created with the Bonfiglioli Vectron inverter configuration utility VPlus using the "Create Ethernet/IP EDS File" function. There are two different methods available. If you are connected to an inverter by PC and service adapter KP232 or Ethernet, an EDS file can be created online. If a vcb-file is available for the inverter, then an EDS file can be created offline.

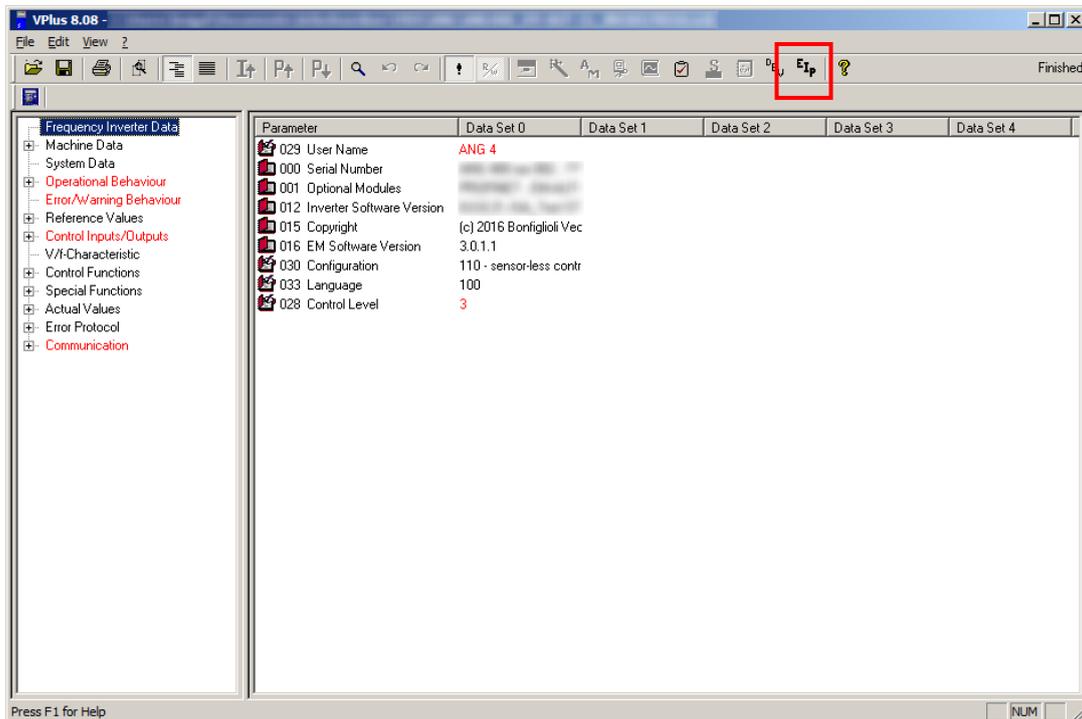


Figure 9-1: Creating EDS files

Online

The "Create Ethernet/IP EDS File" function can be started from the Inverter Manager window:

- select an inverter in the inverter tree
- select menu Edit\Inverter Menu\Create Ethernet/IP EDS File

or

- select an inverter in the inverter tree
- click the "Create Ethernet/IP EDS File" button on the Inverter Manager toolbar

The "Create Ethernet/IP EDS File" function can be started from the Parameter window:

- select menu Edit\Create Ethernet/IP EDS File

or

- click the "Create Ethernet/IP EDS File" button on the Parameter window toolbar

Offline

The "Create Ethernet/IP EDS File" function can be started from the Parameter window:

- select menu Edit\Create Ethernet/IP EDS File

or

- click the "Create Ethernet/IP EDS File" button on the Parameter window toolbar

Irrespective of the method used, a window opens where a default filename for the EDS file is offered and a folder where to store this file. The EDS filename and storage location can be changed by the user.

NOTICE

Risk of component damage!

- When changing an EDS filename, the file extension MUST remain "eds".
- Template files are stored in folder Vplus\Eip. These files must not be deleted.



If there are changes in the EDS file content due to new inverter software, version updates for EDS file templates are provided by the inverter configuration utility and the integrated software update utility (online internet access required).

9.2 Supported configurations

ACU 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 this 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 behavior with respect to *control word / status word* and *modes of operation / modes of operation display* varies in the two different types of configuration.

Standard:

Necessary settings:

- *Configuration 30* ≠ x40.
- *Local/Remote 412* = 2 - Control via Remote-Contacts

The control (Start, Stop, Frequency change over, etc.) is typically carried out via:

- Digital contacts
- Remote contacts via Field bus

Reference values result from the selected configuration. Typical configurations:

Reference speed / Reference frequency:

- Analogue input
- Fixed values from parameters
- Target velocity

Percentage reference value for technology controller or Torque control:

- Analogue input
- Fixed values from parameters

Please refer to chapter "Configurations without motion control" for the control without Positioning functionality.

Positioning via contacts (or remote contacts):

Necessary settings:

- *Configuration 30* = x40.
- *Local/Remote 412* = 0 - Control via Contacts or 2 - Control via 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 configurations:

- 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 - Control via Statemachine

The control (Start, Stop, mode change over, etc.) is carried out via the Control word.

Reference values result from the selected Modes of Operation.

Typical Modes of Operation are:

- Reference speed via target velocity
- Target position

The usage of the Motion Control Interface is described in this manual in Chapters "Motion Control Interface (MCI)" and "Motion control configurations".

9.3 Initialization time

When the frequency inverter is turned on, the automation module must be initialized in addition to the frequency inverter. The initialization can take up to 20 seconds.



Wait until the initialization phase is complete before starting the communication. Module status LED (A) (see 7.1)

9.4 EtherNet/IP™ overview

9.4.1 Class 0x01, Identity Object

Instance	Attribute	Name	Service	Data Type	Value
0	1	Revision	get	UINT	1
1	1	VendorID	get	UINT	1202 Bonfiglioli Vectron MDS GmbH
	2	DeviceType	get	UINT	2 - AC/DC-Drive
	3	ProductCode	get	UINT	value depends on inverter type
	4	Revision	get	Struct USINT/USINT	0x01 0x01
	5	Status	get	WORD	current device state
	6	Serial Number	get	UDINT	
	7	Product Name	get	SHORT_STRIN G	
	8	State	get	USINT	0 – nonexistent 1 – device self testing 2 – Standby 3 – operational 4 – major recoverable fault 5 – major unrecoverable fault
	9	Configuration consistency check	get	UINT	

9.4.2 Class 0xF5, TCP/IP Object

The TCP/IP Interface Object provides the mechanism to configure a device's TCP/IP network interface. The following *Instances* and *Attributes* are supported:

Instance	Attribute	Name	Service	Data Type	Value
0	1	Revision	get	UINT	2
1	1	Status	get	DWORD	see below
	2	Configuration Capability	get	DWORD	0x00000014 DHCP, Configuration settable
	3	Configuration Control	get/set	DWORD	Default = 0 (fixed IP)

Instance	Attribute	Name	Service	Data Type	Value
	4	- Physical Link Object - Path size - Path	get	STRUCT of: UINT Padded EPATH	No. of 16 bit words in Path The path is restricted to one logical class segment and one logical instance segment. The max. size is 12 bytes
	5	Interface Configuration - IP Address - Network Mask - Gateway Address - Name Server (Primary name server) - Name Server 2 (Secondary name server) - Domain Name (Default domain name)	get/set	STRUCT of: UDINT UDINT UDINT UDINT UDINT STRING	see below
	6	Hostname	get/set	STRING	ASCII, max. 64 chars., padded to an even no. of chars. (pad not included in length). A length of 0 indicates no Host Name is configured.
1	8	TTL Value(Time-to-Live value for IPmulticast packets)	get/set ¹⁾	USINT	see below
	9	Mcast Config - Alloc Control(Multicast addressallocation control word) - Reserved - Num Mcast(No. of IP multicast addreses to allocate forEtherNet/IP™) - Mcast Start Addr(Starting multicastaddress from which tobegin allocation)	get/set ¹⁾	STRUCTof: USINTUSINT UINT UDINT	

1) If either TTL Value or Mcast Config is implemented as settable, both must be implemented as settable.

Status

Bit(s)	Name	Value
0-3	Interface Configuration Status	0 – The Interface Configuration attribute has not been configured. 1 – The Interface Configuration attribute contains configuration obtained from BOOTP, DHCP or nonvolatile storage. 2 – The IP address member of the Interface Configuration attribute contains the configuration, obtained from hardware settings (e.g.: pushwheel, thumbwheel, etc.). 3-15 – Reserved for future use.
4	Mcast Pending	Indicates a pending configuration change in the TTL Value and/or Mcast Config attributes. This bit shall be set when either the TTL value or Mcast Config attribute is set, and shall be cleared the next time the device starts.
5	Interface Configuration Pending	Indicates a pending configuration change in the Interface Configuration attribute. This bit shall be 1 (TRUE) when Interface Configuration attribute are set and the device requires a reset in order for the configuration change to take effect (as indicated in the Configuration Capability attribute).

		The intent of the Interface Config Pending bit is to allow client software to detect that a device's IP configuration has changed, but will not take effect until the device is reset.
6	AcdStatus	not supported
7	AcdFault	not supported
8-31	Reserved	0

Configuration Capability

Bit(s)	Name	Value
0	BOOTP Client	1 (TRUE) = device capable of obtaining its network configuration via BOOTP.
1	DNS Client	1 (TRUE) = device capable of resolving host names by querying a DNS server.
2	DHCP Client	1 (TRUE) = device capable of obtaining its network configuration via DHCP.
3	DHCP-DNS Update	Shall be 0, behavior to be defined in a future specification edition.
4	Configuration Settable	1 (TRUE) = Interface Configuration attribute is settable.
5	Hardware Configurable	1 (TRUE) = the IP Address member of the Interface Configuration attribute can be obtained from hardware settings (e.g., pushwheel, thumbwheel, etc.). If this bit is FALSE the Status Instance Attribute (1), Interface Configuration Status field value shall never be 2 (The Interface Configuration attribute contains valid configuration, obtained from hardware settings).
6	Interface Configuration Change Requires Reset	1 (TRUE) = device requires a restart in order for a change to the Interface Configuration attribute to take effect. If this bit is FALSE a change in the Interface Configuration attribute will take effect immediately.
7	AcdCapable	1 (TRUE) = device is ACD capable
8-31	Reserved	0

Configuration Control

Bit(s)	Name	Value
0-3	Configuration Method	0 – The device shall use statically-assigned IP configuration values. 1 – The device shall obtain its configuration values via BOOTP. 2 – The device shall obtain its configuration values via DHCP. 3-15 – Reserved for future use.
4	DNS Enable	If 1 (TRUE), the device shall resolve host names by querying a DNS server.
5-31	Reserved	0

Interface Configuration

Field Name	Data Type	Value
IP Address	UDINT	A value of 0 indicates no IP address has been configured. Otherwise, the IP address shall be set to a valid Class A, B, or C address and shall not be set to the loopback address (127.0.0.1).
Network Mask	UDINT	A value of 0 indicates no Network Mask address has been configured.
Gateway Address	UDINT	A value of 0 indicates no Gateway address has been configured. Otherwise, the Gateway IP address shall be set to a valid Class A, B, or C address and shall not be set to the loopback address (127.0.0.1).
Name Server	UDINT	A value of 0 indicates no Name Server address has been configured. Otherwise, the Name Server address shall be set to a valid Class A, B, or C address.
Name Server 2	UDINT	A value of 0 indicates no Secondary Name Server address has been configured. Otherwise, the Secondary Name Server address shall be set to a valid Class A, B, or C address.
Domain Name	STRING	ASCII, max. 48 chars., padded to an even no. of chars. (pad not included in length). A length of 0 indicates no Domain Name is configured.

TTL Value

The *TTL Value* is the value a device shall use for the IP header Time-to-Live field when sending EtherNet/IP™ packets via IP multicast.

Data Type: USINT Min./Default value is 1, Max. value is 255



Unicast packets shall use the TTL as configured for the TCP/IP stack, and not the TTL Value configured in this attribute.

Mcast Config

The *Mcast Config* attribute contains the configuration of the device's IP multicast addresses to be used for EtherNet/IP™ multicast packets. There are three elements to the *Mcast Config* structure:

- Alloc Control,
- Num Mcast, and
- Mcast Start Addr.

Alloc Control

Value	Definition
0	Multicast addresses shall be generated using the default allocation specified in the CIP EtherNet/IP™ specification. When this value is specified on a set-attribute or set-attributes-all, the values of <i>Num Mcast</i> and <i>Mcast Start Addr</i> in the set-attribute request shall be 0.
1	Multicast addresses shall be allocated according to the values specified in <i>Num Mcast</i> and <i>Mcast Start Addr</i> .
2	Reserved

Num Mcast

Num Mcast is the number of IP multicast addresses allocated. The maximum number of multicast addresses is device specific, but shall not exceed the number of EtherNet/IP™ multicast connections supported by the device.

Mcast Start Addr

Mcast Start Addr is the starting multicast address from which *Num Mcast* addresses are allocated.

9.4.3 Class 0xF6, Ethernet Link Object

The *Ethernet Link* Object maintains link-specific counters and status information for an IEEE 802.3 communications interface.

The following *Instances* and *Attributes* are supported:

Instance	Attribute	Name	Service	Data Type	Value
0	1	Revision	get	UINT	3
1	1	Interface Speed	get	UDINT	Speed in Mbps
	2	Interface Flags (Interface status flags)	get	DWORD	Bitmap of Interface Flags, see below
	3	Physical Address (MAC layer address)	get	USINT[6]	see below

Interface Flags

Bit(s)	Name	Value
0	Link Status	0 – inactive link 1 – active link
1	Half/Full Duplex	0 – half duplex 1 – full duplex Note that if the Link Status flag is 0, then the value of this flag is indeterminate.
2-4	Negotiation Status	Indicates the status of link auto-negotiation 0 – Auto-negotiation in progress 1 – Auto-negotiation and speed detection failed. Using default values for speed and duplex. 2 – Auto negotiation failed but detected speed. Duplex was defaulted. 3 – Successfully negotiated speed and duplex. 4 – Auto-negotiation not attempted. Forced speed and duplex.
5	Manual Setting Requires Reset	0 – The interface can activate changes to link parameters (auto-negotiate, duplex mode, interface speed) automatically. 1 – The device requires a Reset service be issued to its Identity Object in order for the changes to take effect.
6	Local Hardware Fault	0 – The interface detects no local hardware fault. 1 – A local hardware fault is detected.
7-31	Reserved	0

Physical Address

Contains the interface's MAC layer address and is an array of octets.

The recommended display format is "XX-XX-XX-XX-XX-XX-XX", starting with the first octet.

9.4.4 Class 0x04, Assembly Object

The *Assembly Object* provides explicit message access to assembly data. Assembly data can be written/read alternating with I/O data access.

The instance number defines the assembly. Assemblies 20, 21, 22, 24, 70, 71, 72 and 74 have fixed length (number of bytes). Assemblies 100/101 (vendor specific) support dynamic mapping and therefore have variable length (depending on the objects mapped).

Instance	Attribute	Name	Service	Data Type	Value
0	1	Revision	get	UINT	2
20	3	data	set	ARRAY of BYTE	
	4	size	get	UINT	4
21	3	data	set	ARRAY of BYTE	
	4	size	get	UINT	4
22	3	data	set	ARRAY of BYTE	
	4	size	get	UINT	6
24	3	data	set	ARRAY of BYTE	
	4	size	get	UINT	6
100	3	data	set	ARRAY of BYTE	
	4	size	get	UINT	Depends on mapping
70	3	data	get	ARRAY of BYTE	
	4	size	get	UINT	4
71	3	data	get	ARRAY of BYTE	
	4	size	get	UINT	4
72	3	data	get	ARRAY of BYTE	
	4	size	get	UINT	6
74	3	data	get	ARRAY of BYTE	
	4	size	get	UINT	6
101	3	data	get	ARRAY of BYTE	
	4	size	get	UINT	Depends on mapping

9.4.5 Class 0x0F, Parameter Object

The *Parameter Object* refers to EtherNet/IP™ Parameters that can be read/written. These parameters are used for different settings of the inverter e.g. setting the input/output assembly selection.

Instance	Attribute	Name	Service	Data Type	Value
0	2	Max Instance	get	UINT	129
	8	Parameter Class Descriptor	get	WORD	5 Bit 0 = 1: supports parameter instances Bit 2 = 1: must do nonvolatile storage save command
	9	Configuration assembly Instances	get	UINT	0 – configuration assembly is not supported

Available Services for Class 0x0F / Instance 0 / Attribute don't care:

- *Reset* Resets all parameters to the factory default
- *Restore* Restores all parameter values from non-volatile memory
- *Save* Saves all parameter values to non-volatile memory

9.4.5.1 Input Assembly Selection, Parameter 1

Instance	Attribute	Name	Service	Data Type	Value
1	1	Param1 InputAssemblySelection	get/set	UINT	70 – Assembly 70 71 – Assembly 71 72 – Assembly 72 74 – Assembly 74 101 – Assembly 101

Parameter **1** "*InputAssemblySelection*" defines the used input assembly. The object content of the different assemblies is described in Chapter "I/O Assemblies".

Input assemblies 70/71/72/74 have fixed objects and length.

Input assembly 101 is a vendor specific assembly with variable mapping and up to 12 objects / 24 bytes provided by CANopen definitions. For a detailed description please refer to Chapter "Vendor Specific Assemblies 100/101".



This parameter is only settable if no I/O-Connection is established.

9.4.5.2 Input Assembly Length, Parameter 2

Instance	Attribute	Name	Service	Data Type	Value
2	1	Param2 InputAssemblyLength	get	UINT	Depends on selected input assembly (and variable mapping assembly 101) 1 ... 24 bytes

Parameter 2 "*InputAssemblyLength*" contains the length of the currently set input assembly.

9.4.5.3 Input Assembly Mapping for Vendor Specific Input Assembly 101, Parameters 3 to 14

Instance	Attribute	Name	Service	Data Type	Value
3	1	Param3 InputObject1	get/set	UINT	See list
4	1	Param4 InputObject2	get/set	UINT	
5	1	Param5 InputObject3	get/set	UINT	
6	1	Param6 InputObject4	get/set	UINT	
7	1	Param7 InputObject5	get/set	UINT	
8	1	Param8 InputObject6	get/set	UINT	
9	1	Param9 InputObject7	get/set	UINT	
10	1	Param10 InputObject8	get/set	UINT	
11	1	Param11 InputObject9	get/set	UINT	
12	1	Param12 InputObject10	get/set	UINT	
13	1	Param13 InputObject11	get/set	UINT	
14	1	Param14 InputObject12	get/set	UINT	

Parameter 3 *InputObject1* to Parameter 14 *InputObject12*, define the content of vendor specific Input Assembly 101. A choice list is available for these parameters.



These parameters are only settable if no I/O-connection is established or the connection is not to Assembly 101.

Choice list for Parameters 3 to 14:

- 0 "- no object mapped here",
- 41 "0x3001 (2B) Digital In actual values",
- 42 "0x3002 (2B) Digital Out actual values",
- 44 "0x3004 (2B) Boolean Mux",
- 47 "0x3007 (2B) Percentage actual value",
- 48 "0x3008 (2B) Percentage actual value2",
- 49 "0x3011 (2B) Act. value word1 (P.1415)",
- 50 "0x3012 (2B) Act. value word2 (P.1416)",

- 51 "0x3021 (4B) Act. value long1 (P.1417)",
- 52 "0x3022 (4B) Act. value long2 (P.1418)",
- 76 "0x5FF0 (1B) active motion block (ttr)",
- 77 "0x5FF1 (1B) motion block to resume (ttr)",
- 81 "0x6041 (2B) statusword",
- 83 "0x6043 (2B) velocity demand",
- 84 "0x6044 (2B) control effort",
- 94 "0x6061 (1B) modes of operation display",
- 95 "0x6064 (4B) position actual value",
- 100 "0x606C (4B) velocity actual value",
- 106 "0x6077 (2B) Torque actual value",
- 107 "0x6078 (2B) Current actual value",
- 108 "0x6079 (4B) DC link voltage",
- 126 "0x60F4 (4B) following error value";

This selection above lists the CANopen objects which can be mapped to Input Assembly 101. The values in brackets, e.g. (2B) or (4B), define the number of bytes of the corresponding object. The maximum number of bytes for input assembly is 24.

For a detailed description please refer to Chapter "Vendor Specific Assemblies 100/101".

9.4.5.4 Output Assembly Selection, Parameter 15

Instance	Attribute	Name	Service	Data Type	Value
15	1	Param15 OutputAssemblySelection	get/set	UINT	20 – Assembly 20 21 – Assembly 21 22 – Assembly 22 24 – Assembly 24 100 – Assembly 100

Parameter **15** "*OutputAssemblySelection*" defines the used output assembly. The object content of the different assemblies is described in Chapter "I/O Assemblies".

Output assemblies 20/21/22/24 have fixed objects and length.

Output assembly 100 is a vendor specific assembly with variable mapping and up to 12 objects / 24 bytes. For a detailed description please refer to Chapter "Vendor Specific Assemblies 100/101".



These parameters are only settable if no I/O-connection is established.

9.4.5.5 Output Assembly Length, Parameter 16

Instance	Attribute	Name	Service	Data Type	Value
16	1	Param16 OutputAssemblyLength	get	UINT	Depends on selected input assembly (and variable mapping assembly 100) 1 ... 24 bytes

Parameter 16 "*OutputAssemblyLength*" contains the length of the currently set output assembly.

9.4.5.6 Output Assembly Mapping for Vendor specific Output Assembly 100, Parameters 17 to 28

Instance	Attribute	Name	Service	Data Type	Value
17	1	Param17 OutputObject1	get/set	UINT	See list
18	1	Param18 OutputObject2	get/set	UINT	
19	1	Param19 InputObject3	get/set	UINT	
20	1	Param20 OutputObject4	get/set	UINT	

21	1	Param21 InputObject5	get/set	UINT	
22	1	Param22 OutputObject6	get/set	UINT	
23	1	Param23 OutputObject7	get/set	UINT	
24	1	Param24 OutputObject8	get/set	UINT	
25	1	Param25 OutputObject9	get/set	UINT	
26	1	Param126 OutputObject10	get/set	UINT	
27	1	Param127 OutputObject11	get/set	UINT	
28	1	Param28 OutputObject12	get/set	UINT	

Parameter **17** *OutputObject1* to Parameter **28** *OutputObject12* define the content of vendor-specific Output Assembly 100. A choice list is available for these parameters.



These parameters are only settable if no I/O-connection is established or the connection is not to Assembly 100.

Choice list for Parameters **17** to **28**:

- 0 "- no object mapped here",
- 43 "0x3003 (1B) Digital Out set values",
- 45 "0x3005 (2B) Boolean Demux",
- 46 "0x3006 (2B) Percentage set value",
- 53 "0x3111 (2B) Ref. value word1 (S.762)",
- 54 "0x3112 (2B) Ref. value word2 (S.763)",
- 55 "0x3121 (4B) Ref. value long1 (S.764)",
- 56 "0x3122 (4B) Ref. value long2 (S.765)",
- 57 "0x5F10_1 (2B) Numerator",
- 58 "0x5F10_2 (2B) Denominator",
- 80 "0x6040 (2B) controlword",
- 82 "0x6042 (2B) target velocity",
- 93 "0x6060 (1B) modes of operation",
- 105 "0x6071 (2B) Target torque",
- 109 "0x607A (4B) target position",
- 111 "0x6081 (4B) profile velocity",
- 112 "0x6083 (4B) profile acceleration",
- 113 "0x6084 (4B) profile deceleration",
- 114 "0x6085 (4B) quick stop deceleration",
- 128 "0x60FF (4B) target velocity";

This selection defines CANopen objects to be mapped to Output Assembly 100. The values in brackets, e.g. (2B) or (4B), define the number of bytes of the corresponding object. The maximum number of bytes for output assembly is 24.

For a detailed description please refer to Chapter "Vendor Specific Assemblies 100/101".

9.4.5.7 Reserved Parameters 29 - 32

Parameters **29** to **32** have no function and are reserved for future use.

9.4.5.8 Ethernet/IP Abort Behavior, Parameter 33

Instance	Attribute	Name	Service	Data Type	Value
33	1	Param33 Ethernet/IP Abort Behavior	get	USINT	See list

Parameter **33** "*Ethernet/IP Abort Behavior*" defines inverter behavior in the event of bus errors.



This has the same effect as writing Inverter Parameter **388** "*Error Behavior*".

Choice list for parameter **33**:

- 0 no reaction
- 1 fault
- 2 stop (without fault message)
- 3 emergency stop (without fault message)
- 4 stop + fault
- 5 emergency stop + fault



Settings 2, 3, 4 and 5 are only executed as described by the inverter, if Parameter *Local/Remote* **412** is set to "1 - Remote State Machine". With all other settings of **P.412**, the settings 2, 3, 4 and 5 trigger a fault reaction as in choice 1.

9.4.5.9 Change of State (COS) Mask, Parameters 34 to 39

Instance	Attribute	Name	Service	Data Type	Value
34	1	Param34 COS Mask 0	get/set	DWORD	0x00000000 to 0xFFFFFFFF (Default = 0xFFFFFFFF)
35	1	Param35 COS Mask 1	get/set	DWORD	
36	1	Param36 COS Mask 2	get/set	DWORD	
37	1	Param37 COS Mask 3	get/set	DWORD	
38	1	Param38 COS Mask 4	get/set	DWORD	
39	1	Param39 COS Mask 5	get/set	DWORD	

Parameters **34** "*COS Mask 0*" to **39** "*COS Mask 5*" define a bit mask for the COS function. These can be used to exclude small fluctuations of actual values (i.e. small speed changes) in the evaluation of whether a value has changed, and therefore reduce the number of transmissions.

In the case of input data communication type *Change-Of-State* (COS), the *COS Mask* defines the bits of the *Input Assembly* that are taken into account for the COS function. Every *Input Assembly* consists of a specific number of bytes. Every bit in the *Input Assembly* bytes corresponds to one bit in the *COS Mask*. If the corresponding bit in the *COS Mask* is set to "1", then this bit in the *Input Assembly* is used for the COS function.

If the corresponding bit in the *COS Mask* is set to "0", then this bit in the *Input Assembly* has no effect for the COS function. A change of this bit in the *Input Assembly* does not trigger the sending of the *Input Assembly*.

The maximum length of the *Input Assembly* is 24 bytes = 192 bits. The six *COS Masks* have 32 bits each and also 192 bits in total.

The relationship between *COS Mask* and *Input Assembly* is defined as:

Input Assembly Byte No.	COS Mask	Instance (Parameter)
0 – 3	COS Mask 0	34
4 – 7	COS Mask 1	35
8 – 11	COS Mask 2	36
12 – 15	COS Mask 3	37
16 – 19	COS Mask 4	38
20 – 23	COS Mask 5	39

Example:

Input Assembly byte **0** and low byte of *COS Mask 0*.

COS Mask 0 / Byte 0 / Bit 0 – 7							
Bit	7	6	5	4	3	2	1
	1	1	1	1	1	0	0
	n	n	n	n	n	x	X
Bit	7	6	5	4	3	2	1
Input Assembly / Byte 0 / Bit 0 – 7							

The low byte of *COS Mask 0* is set to 0xF8. Any change in bits 0 – 2 of the *Input Assembly* has no effect. A change in bits 3 – 7 triggers the sending of the *Input Assembly*.

9.4.5.10 Reserved Parameter 40

Instance	Attribute	Name	Service	Data Type	Value
40	1	Param40	-	-	reserved for future use

Parameter **40** has no function and is reserved for future use.

9.4.5.11 CANopen Objects, Parameters 41 – 129

Parameters **41** to **129** are CANopen objects. Some of these objects can be mapped to Input Assembly 101 or Output Assembly 100.

See Chapter "Input Assembly Mapping for Vendor Specific Input Assembly 101, Parameters 3 to 14" and Chapter "Output Assembly Mapping for Vendor specific Output Assembly 100, Parameters 17 to 28".

The use of these parameters is described in Chapter 9.9 "CANopen objects".

All CANopen objects can be accessed by explicit message.

The use of CANopen objects with vendor specific Input Assembly 101 and Output Assembly 100 provides CANopen functionality (e.g. motion control functions) to the EtherNet/IP™ system.



All of the CANopen functions are available, **with the exception of** motion control "*interpolated position mode*" and "*cyclic synchronous position mode*".

"*Interpolated position mode*" requires equidistant timing for output/input assemblies, which is not supported by EtherNet/IP™.

Instance	Attribute	Name	Service	Data Type
41	1	Param41 CANopen_3001 Digital in actual value	get	WORD
42	1	Param42 CANopen_3002 Digital out actual value	get	WORD
43	1	Param43 CANopen_3003 Digital out set values	get/set	USINT
44	1	Param44 CANopen_3004 Boolean Mux	get	WORD
45	1	Param45 CANopen_3005 Boolean Demux	get/set	WORD
46	1	Param46 CANopen_3006 Percentage set value	get/set	INT
47	1	Param47 CANopen_3007 Percentage actual value	get	INT
48	1	Param48 CANopen_3008 Percentage actual value2	get	INT
49	1	Param49 CANopen_3011 Act. Value Word 1 p.1415	get	INT
50	1	Param50 CANopen_3012 Act. Value Word 2 p.1416	get	INT
51	1	Param51 CANopen_3021 Act. Value Long 1 p.1417	get	DINT
52	1	Param52 CANopen_3022 Act. Value Long 2 p.1418	get	DINT
53	1	Param53 CANopen_3111 Ref. Value Word 1 s.762	get/set	INT

Instance	Attribute	Name	Service	Data Type
54	1	Param54 CANopen_3112 Ref. Value Word 2 s.763	get/set	INT
55	1	Param55 CANopen_3121 Ref. Value Long 1 s.764	get/set	DINT
56	1	Param56 CANopen_3122 Ref. Value Long 2 s.765	get/set	DINT
57	1	Param57 CANopen_5F10 Gear Factor Numerator	get/set	INT
58	1	Param58 CANopen_5F10 Gear Factor Denominator	get/set	UINT
59	1	Param59 CANopen_5F10 Resync on gear factor change	get/set	INT16
60	1	Param60 CANopen_5F11_1 Phasing 1 Offset	get/set	DINT
61	1	Param61 CANopen_5F11_2 Phasing 1 Speed	get/set	UDINT
62	1	Param62 CANopen_5F11_3 Phasing 1 Acceleration	get/set	UDINT
63	1	Param63 CANopen_5F12_1 Phasing 2 Offset	get/set	DINT
64	1	Param64 CANopen_5F12_2 Phasing 2 Speed	get/set	UDINT
65	1	Param65 CANopen_5F12_3 Phasing 2 Acceleration	get/set	UDINT
66	1	Param66 CANopen_5F13_1 Phasing 3 Offset	get/set	DINT
67	1	Param67 CANopen_5F13_2 Phasing 3 Speed	get/set	UDINT
68	1	Param68 CANopen_5F13_3 Phasing 3 Acceleration	get/set	UDINT
69	1	Param69 CANopen_5F14_1 Phasing 4 Offset	get/set	DINT
70	1	Param70 CANopen_5F14_2 Phasing 4 Speed	get/set	UDINT
71	1	Param71 CANopen_5F14_3 Phasing 4 Acceleration	get/set	UDINT
72	1	Param72 CANopen_5F15 In Gear Threshold	get/set	UDINT
73	1	Param73 CANopen_5F16 In Gear Time	get/set	UINT
74	1	Param74 CANopen_5F17_1 Position Controller Time constant	get/set	INT
75	1	Param75 CANopen_5F17_2 Position Controller Limitation	get/set	DINT
76	1	Param76 CANopen_5FF0 active motion block	get	SINT
77	1	Param77 CANopen_5FF1 motion block to resume	get	SINT
78	1	Param78 CANopen_6007 abort connection option code	get/set	INT
79	1	Param79 CANopen_603F error code	get	UINT
80	1	Param80 CANopen_6040 controlword	get/set	WORD
81	1	Param81 CANopen_6041 statusword	get	WORD
82	1	Param82 CANopen_6042 target velocity	get/set	INT
83	1	Param83 CANopen_6043 velocity demand	get	INT
84	1	Param84 CANopen_6044 control effort	get	INT
85	1	Param85 CANopen_6046_1 velocity min amount	get/set	UDINT
86	1	Param86	get/set	UDINT

Instance	Attribute	Name	Service	Data Type
		CANopen_6046_2 velocity max amount		
87	1	Param87 CANopen_6048_1 velocity acceleration delta speed	get/set	UDINT
88	1	Param88 CANopen_6048_2 velocity acceleration delta time	get	UINT
89	1	Param89 CANopen_6049_1 velocity deceleration delta speed	get/set	UDINT
90	1	Param90 CANopen_6049_2 velocity deceleration delta time	get/set	UINT
91	1	Param91 CANopen_604A_1 velocity quick stop delta speed	get/set	UDINT
92	1	Param92 CANopen_604A_2 velocity quick stop delta time	get/set	UINT
93	1	Param93 CANopen_6060 modes of operation	get/set	SINT
94	1	Param94 CANopen_6061 modes of operation display	get	SINT
95	1	Param95 CANopen_6064 position actual value	get	DINT
96	1	Param96 CANopen_6065 following error window	get/set	UDINT
97	1	Param97 CANopen_6066 following error time out	get/set	UINT
98	1	Param98 CANopen_6067 position window	get/set	UDINT
99	1	Param99 CANopen_ position window time	get/set	UINT
100	1	Param100 CANopen_606C velocity actual value	get	DINT
101	1	Param101 CANopen_606D velocity window	get/set	UINT
102	1	Param102 CANopen_606E velocity window time	get/set	UINT
103	1	Param103 CANopen_606F velocity threshold	get/set	UINT
104	1	Param104 CANopen_6070 velocity threshold time	get/set	UINT
105	1	Param105 CANopen_6071 target torque	get/set	INT
106	1	Param106 CANopen_6077 torque actual value	get	SINT
107	1	Param107 CANopen_6078 current actual value	get	INT
108	1	Param108 CANopen_6079 DC link voltage	get	UDINT
109	1	Param109 CANopen_607A target position	get/set	DINT
110	1	Param110 CANopen_607C home offset	get/set	DINT
111	1	Param111 CANopen_6081 profile velocity	get/set	UDINT
112	1	Param112 CANopen_6083 profile acceleration	get/set	UDINT
113	1	Param113 CANopen_6084 profile deceleration	get/set	UDINT
114	1	Param114 CANopen_6085 quick stop deceleration	get/set	UDINT
115	1	Param115 CANopen_6086 motion profile type	get/set	INT
116	1	Param116 CANopen_6087 torque slope	get/set	UDINT

Instance	Attribute	Name	Service	Data Type
117	1	Param117 CANopen_6091_1 gear ratio motor revolutions	get/set	UDINT
118	1	Param118 CANopen_6091_2 gear ratio shaft revolutions	get/set	UDINT
119	1	Param119 CANopen_6092_1 feed constant feed	get/set	UDINT
120	1	Param120 CANopen_6092_2 feed constant shaft revolutions	get/set	UDINT
121	1	Param121 CANopen_6098 homing method	get/set	SINT
122	1	Param122 CANopen_6099_1 homing speeds speed during search for switch	get/set	UDINT
123	1	Param123 CANopen_6099_2 homing speeds speed during search for zero	get/set	UDINT
124	1	Param124 CANopen_609A homing acceleration	get/set	UDINT
126	1	Param126 CANopen_60F4 following error value	get	DINT
127	1	Param127 CANopen_60F8 max slippage	get/set	DINT
128	1	Param128 CANopen_60FF target velocity	get/set	DINT
129	1	Param128 CANopen_5F18 M/S-Synchronization offset	get/set	DINT

9.4.6 Class 0x28, Motor Data Object

The values of the *Motor Data Object* attributes are read from internal inverter parameters. They are **always** read from data set 1, even if data sets **1** to **4** have different values.

Instance	Attribute	Name	Service	Data Type	Value
1	3	MotorType	get	USINT	0 – Non-Standard 3 – PM Synchronous 7 – Squirrel Cage Induction
	6	RatedCurrent	get	UINT	xxx [100 mA]
	7	RatedVoltage	get	UINT	xxx [V]
	8	RatedPower	get	UDINT	xxx [W]
	9	RatedFrequency	get	UINT	xxx [Hz]
	12	PoleCount	get	UINT	xxx
	15	BaseSpeed	get	UINT	xxx [RPM]

9.4.7 Class 0x29, Control Supervisor Object

All attributes of the *Control Supervisor Object* can be read / written by explicit message. They are assembled in various pre-defined assemblies for I/O data exchange. Please refer to Chapter "I/O Assemblies" for correct handling of attributes and necessary basic settings of inverter parameters.

Instance	Attribute	Name	Service	Data Type	Value
1	3	Run1	get/set	BOOL	1 – Run Forward (=Clockwise ?)
	4	Run2	get/set	BOOL	1 – Run Reverse (=Anticlockwise ?)
	5	NetCtrl	get/set	BOOL	0 – Local control 1 – Network control
	6	State	get	USINT	1 – Startup

					2 – Not_Ready 3 – Ready 4 – Enabled 5 – Stopping 6 – Fault_Stop 7 - Faulted
	7	Running1	get	BOOL	1 – Enabled and Run1 (= Clockwise)
	8	Running2	get	BOOL	1 – Enabled and Run2 (= Anticlockwise)
	9	Ready	get	BOOL	1 – Ready or Enabled or Stopping
	10	Faulted	get	BOOL	1 – Fault Occurred
	11	Warning	get	BOOL	1 – Warning
	12	FaultRst	get/set	BOOL	0 → 1 Fault Reset
	15	CtrlFromNet	get	BOOL	0 – Control is local 1 – Control is from network

9.4.8 Class 0x2A, AC/DC-Drive Object

All attributes of the *AC/DC-Drive Object* can be read / written by explicit message. They are assembled in various pre-defined assemblies for I/O data exchange. Please refer to Chapter 9.7 "I/O Assemblies" for correct handling of attributes and necessary basic settings of inverter parameters.

Instance	Attribute	Name	Service	Data Type	Value
1	3	AtReference	get	BOOL	1 – Drive actual at reference
	4	NetRef	get/set	BOOL	0 – Set reference not EIP control (local reference) 1 – Set Reference at EIP control (PLC)
	6	DriveMode	get	USINT	0 – Vendor specific 1 – Open Loop Speed 2 – Closed Loop Speed 3 – Torque Control 4 – Process Control 5 – Position Control
	7	SpeedActual	get	INT	xxxx [RPM]
	8	SpeedRef	get/set	INT	xxxx [RPM]
	11	TorqueActual	get	INT	xxxx [Nm]
	12	TorqueRef	get/set	INT	xxxx [Nm]
	13	ProcessActual	get	INT	xxxx [%]
	14	ProcessRef	get/set	INT	xxxx [%]
	29	RefFromNet	get	BOOL	0 – Local torque/speed reference 1 – Network torque/speed reference

EIP = EtherNet/IP™

9.5 Operating behavior in the case of bus connection failure

The operating behavior in the case of failure of the EtherNet/IP™ systems can be parameterized. The required behavior is set via parameter *Bus Error Behaviour* **388**.

Bus Error Behaviour 388	Function
0 - no reaction	Operating point is maintained.
1 - Error	"Fault" status will be activated immediately. Factory setting.
2 - Stop	Device state machine processes control command "Disable voltage" and switches to "switch on disabled" status.
3 - Quick stop	Control command "Quick stop" and switch to "switch on disabled" status.
4 - Shutdown + Error	Control command "Disable operation" and switch to "Error" status once the drive has been shut down.

5 - Quick stop + Error	Control command "Quick stop" and switch to "Error" status once the drive has been shut down.
------------------------	--



The parameter settings *Bus Error Behaviour* **388** = 2...5 are evaluated depending on the setting of the parameter *Local/Remote* **412**.
 The described reactions are only available whenever parameter *LocalRemote* **412** is set to 1 (remote state machine). Otherwise setting parameter *Error Behavior* **388** to values 2 ... 5 always results in the device state machine changing to state "fault" immediately.

The error and warning behavior of the frequency inverter can be parameterized in various ways. For details about possible faults, refer to Chapter 18.4 "Error messages".

9.6 Access to Inverter Parameters (Classes 0x64 ... 0x73)

All inverter parameters are accessible by explicit message. Inverter parameters are addressed by parameter number (**0 ... 1599**) and data set number (**0, 1 ... 9**).

The necessary values for class/instance/attribute for R/W access by explicit message are calculated by the algorithm:

- class = (parameter number / 100) + 100
- attribute = (parameter number MODULO 100) + 1
- instance = data set + 1

Example: P.520 fixed percentage 1, data set 3 (data type = int, 2 bytes)

- class = (520 / 100) + 100 = 5 + 100 = 105
- attribute = (520 MODULO 100) + 1 = 20 + 1 = 21
- instance = 3 + 1 = 4

Refer to the parameter list of the respective configuration in the standard operating instructions to see the parameters that can be set. The parameter list states whether a parameter is data set change-over capable (data set = 1 to 4) or only exists once (data set = 0).

The parameter list also provides information about the display format of a parameter and its type (int/uint/long/string). For values with decimal places, the decimal point is not transmitted.

Typically, parameters are addressed via data set = **0, 1** to **4**. Written values are thus stored to the controller RAM and automatically to the EEPROM. If values are written frequently, no entry shall be made into the EEPROM, as the latter only possesses a limited number of admissible writing cycles (approx. 1 million cycles).

NOTICE

Risk of component damage!

If a maximum number of write cycles for the EEPROM is exceeded, it will be destroyed. The values are entered automatically in the EEPROM of the controller. However, only a limited number of write cycles is permissible for the EEPROM (approx. 1 million cycles).

- If cyclic writing of data is required, use the RAM.

This mechanism is activated by the target data set being increased by five in the specification of the data set.

EEPROM	RAM
Entry into data set 0	Data set = 5
Entry into data set 1	Data set = 6
Entry into data set 2	Data set = 7
Entry into data set 3	Data set = 8
Entry into data set 4	Data set = 9

If the values of a data set change-over capable parameter are to be set to the same value in all data sets, the parameter can be written via data set 0 (EEPROM) or 5 (RAM).

9.7 I/O Assemblies

I/O-Assemblies are pre-defined object assemblies for I/O data with defined content. The assemblies to be used are selected by:

- Parameter 1 (Param1) "*InputAssemblySelection*" for input values to the PLC
 - o choice list for input assemblies: 70 (default)/71/72/74/101
- Parameter 15 (Param15) "*OutputAssemblySelection*" for output values from the PLC
 - o choice list for output assemblies 20 (default)/21/22/24/100

The selection can be made by explicit message, or with the VPlus control software.



These parameters are only settable if no I/O Connection is established.

VPlus

The assembly selection is carried out by writing the following inverter parameters:

- *Input Assembly Selection* **P.1445**
- *Output Assembly Selection* **P.1447**

The length of the currently selected assembly can be read back from the inverter with:

- *Input Assembly Length* **P.1446**
- *Output Assembly Length* **P.1448**

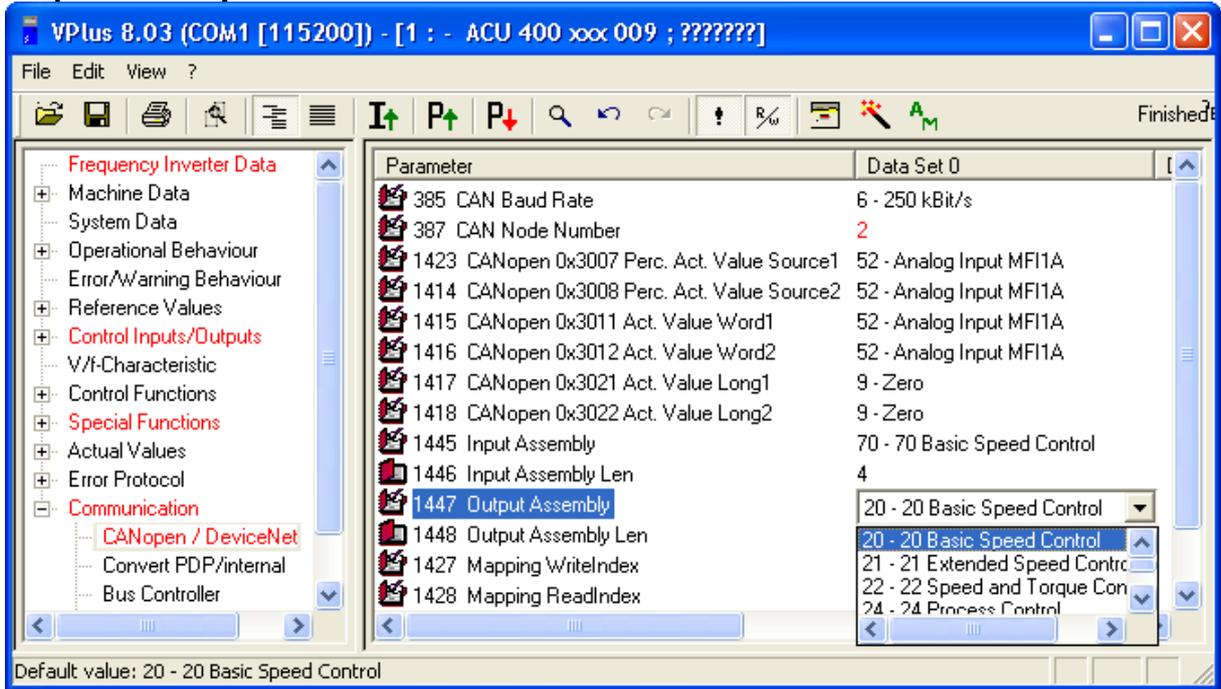
Input Assembly

The screenshot shows the VPlus 8.03 software interface. The title bar reads "VPlus 8.03 (COM1 [115200]) - [1 : - ACU 400 xxx 009 ; ???????]". The left sidebar shows a tree view with categories like "Frequency Inverter Data", "Machine Data", "System Data", "Operational Behaviour", "Error/Warning Behaviour", "Reference Values", "Control Inputs/Outputs", "V/f-Characteristic", "Control Functions", "Special Functions", "Actual Values", "Error Protocol", and "Communication". The "Communication" category is expanded to show "CANopen / DeviceNet". The main window displays a table of parameters for "Data Set 0".

Parameter	Data Set 0
385 CAN Baud Rate	6 - 250 kBit/s
387 CAN Node Number	2
1423 CANopen 0x3007 Perc. Act. Value Source1	52 - Analog Input MF1A
1414 CANopen 0x3008 Perc. Act. Value Source2	52 - Analog Input MF1A
1415 CANopen 0x3011 Act. Value Word1	52 - Analog Input MF1A
1416 CANopen 0x3012 Act. Value Word2	52 - Analog Input MF1A
1417 CANopen 0x3021 Act. Value Long1	9 - Zero
1418 CANopen 0x3022 Act. Value Long2	9 - Zero
1445 Input Assembly	70 - 70 Basic Speed Control
1446 Input Assembly Len	70 - 70 Basic Speed Control
1447 Output Assembly	71 - 71 Extended Speed Contr
1448 Output Assembly Len	72 - 72 Speed and Torque Con
	74 - 74 Process Control

Default value: 70 - 70 Basic Speed Control

Output Assembly



9.7.1 CIP Assemblies 20/21/22/24 + 70/71/72/74

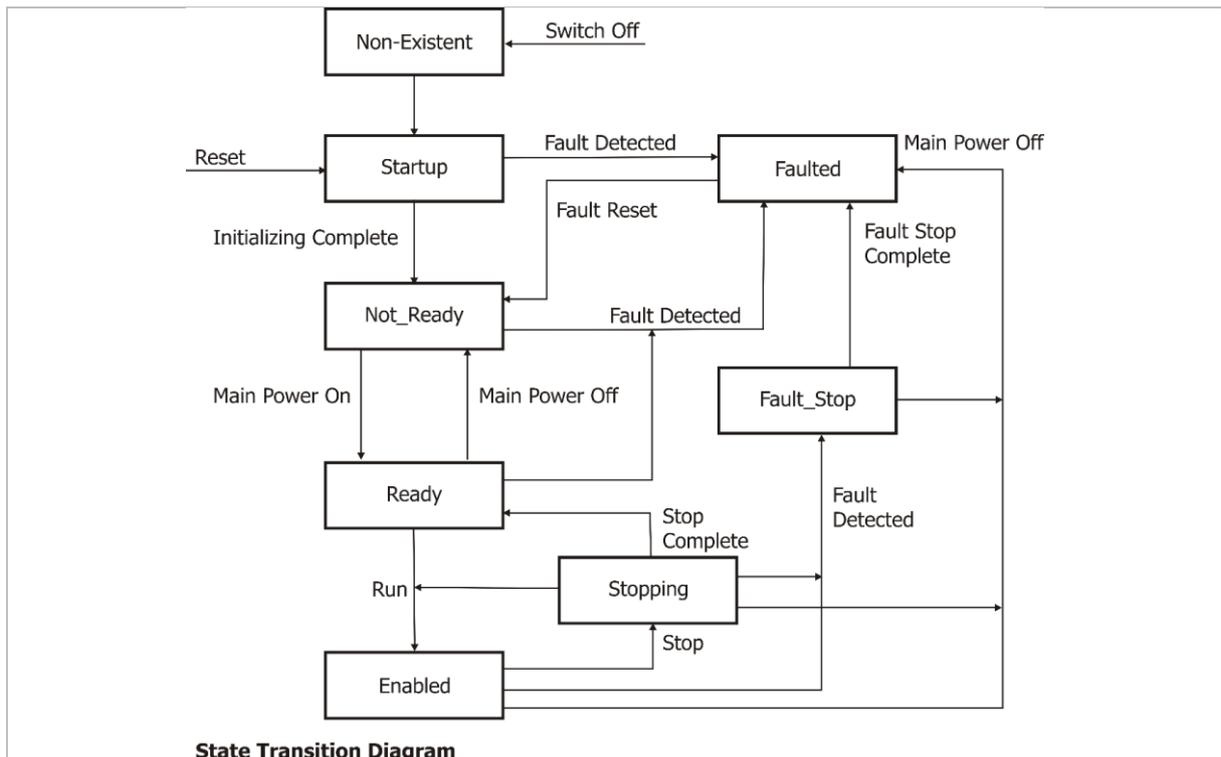


The described functionality of CIP Assemblies requires the following inverter settings:

- Parameter *Local/Remote* **412** = 1 (Remote State Machine)

Parameter **412** is automatically set to 1 whenever *NetCtrl* (Class 0x29, Instance 1, Attribute 5) is set to 1.

- Digital input *STOA* and *STOB* set to *high* (+24V)
- Digital input *Start Clockwise* OR *Start Anticlockwise* set to *high* (+24V) or directly set to *On* (**P.68/P.69**)



The actual state of the drive is displayed in the *Control Supervisor Object* (Class 0x29, Instance 1, Attribute 6 "state").

- 1 - Startup
- 2 - Not_Ready
- 3 - Ready
- 4 - Enabled
- 5 - Stopping
- 6 - Fault_Stop
- 7 - Faulted

9.7.1.1 Output Assembly 20

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	Fault Reset	-	Run Forward
1	-	-	-	-	-	-	-	-
2	Speed Reference (low byte)							
3	Speed Reference (high byte)							

- unused, no function

Run Forward = 1 drive runs clockwise

Fault Reset = 0 → 1, acknowledge fault

Speed Reference = nnnn [RPM]

(written to inverter speed reference Line Setpoint, see **P.434**)

Speed Reference is written to the inverter reference frequency (*Reference Frequency RAM*). The reference frequency is calculated from *Speed Reference* [RPM] and Parameter *No. of Pole Pairs 373* in the active data set.

The data type of *Speed Reference* is INT. Setting *Speed Reference* to negative values causes the drive to run anticlockwise on *Run Forward* set to 1.



The EtherNet/IP™ objects *NetCtrl* (Class 0x29, Instance 1, Attribute 5) and *NetRef* (Class 0x2A, Instance 1, Attribute 4) **must** be set by explicit message from the PLC Program in order to set the inverter to the appropriate control mode.

NetCtrl sets Inverter Parameter *Local/Remote 412* = 1 "Control via Statemachine" (RAM only)

NetRef sets Inverter Parameter *Ramp Setpoint 434* = 2 "Line Setpoint" (RAM only)

This can be achieved by sending explicit messages to these objects individually, or setting both objects together by one explicit message, writing to *Class 4* (Assembly Object), *Instance 21* (Assembly 21), *Attribute 3* (Assembly Data).

For a description and example of how to send explicit messages from the PLC Program, please refer to Chapter Sending Explicit Messages from the PLC Program.

Ramp Setpoint 434	Function
1 - Internal setpoint	
2 - Line setpoint	
3 - Internal+Line setpoint	default

9.7.1.2 Output Assembly 21

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	Net Ref	Net Ctrl	-	-	Fault Reset	Run Reverse	Run Forward
1	-	-	-	-	-	-	-	-
2	Speed Reference (low byte)							
3	Speed Reference (high byte)							

- unused, no function

Run Forward = 1 drive runs clockwise

Run Reverse = 1 drive runs anticlockwise

Setting both objects, *Run Forward* and *Run Reverse*, to 1 shall have no effect.

Fault Reset = 0 → 1, acknowledge fault

Speed Reference = nnnn [RPM]

(written to inverter speed reference Line Setpoint, see **P.434**)

Speed Reference is written to the inverter reference frequency (*Reference Frequency RAM*). The reference frequency is calculated from *Speed Reference* [RPM] and Parameter *No. of Pole Pairs* **373** in the active data set.

The data type of *Speed Reference* is INT. Setting *Speed Reference* to negative values causes the drive to run anticlockwise whenever *Run Forward* is set to 1. Otherwise, setting *Speed Reference* to negative values causes the drive to run clockwise whenever *Run Reverse* is set to 1.

- *Net Ctrl* = 1 inverter controlled by net
- *Net Ctrl* = 0 inverter controlled by local control
- *Net Ctrl* = 1 sets Parameter *LocalRemote* **412** to 1 - Remote Statemachine in all data sets.
- *Net Ctrl* = 0 restores Parameter *LocalRemote* **412** to values from before *Net Ctrl* = 0 → 1
- *Net Ref* = 1 speed reference from net
- *Net Ref* = 0 speed reference from local control
- *Net Ref* = 1 sets Parameter *Ramp Setpoint* **434** to 2 - Line Setpoint in all data sets.
- *Net Ref* = 0 restores Parameter *Ramp Setpoint* **434** to values from before *Net Ref* 0 → 1.



The EtherNet/IP™ objects *NetCtrl* (Class 0x29, Instance 1, Attribute 5) and *NetRef* (Class 0x2A, Instance 1, Attribute 4) **must** be set by explicit message from the PLC Program in order to set the inverter to the appropriate control mode.

NetCtrl sets Inverter Parameter *Local/Remote* **412** = 1 "Control via Statemachine" (RAM only)

NetRef sets Inverter Parameter *Ramp Setpoint* **434** = 2 "Line Setpoint" (RAM only)

This can be achieved by sending explicit messages to these objects individually, or setting both objects together by one explicit message, writing to *Class 4* (Assembly Object), *Instance 21* (Assembly 21), *Attribute 3* (Assembly Data).

For a description and example of how to send explicit messages from the PLC Program, please refer to Chapter Sending Explicit Messages from the PLC Program.

Ramp Setpoint 434	Function
1 - Internal setpoint	
2 - Line setpoint	
3 - Internal+Line setpoint	default

9.7.1.3 Output Assembly 22

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	Fault Reset	-	Run Forward
1	-	-	-	-	-	-	-	-
2	Speed Reference (low byte)							
3	Speed Reference (high byte)							
4	Torque Reference (low byte)							
5	Torque Reference (high byte)							

- unused, no function

Run Forward = 1 drive runs clockwise

Fault Reset = 0 → 1, acknowledge fault

Speed Reference = nnnn [RPM]

(written to inverter speed reference Line Setpoint, see **P.434**)

Speed Reference is written to the inverter reference frequency (*Reference Frequency RAM*). The reference frequency is calculated from *Speed Reference* [RPM] and Parameter *No. of Pole Pairs* **373** in the active data set.

The data type of *Speed Reference* is INT. Setting *Speed Reference* to negative values causes the drive to run anticlockwise whenever *Run Forward* is set to 1.

Torque Reference = nnnn [Nm]

Torque Reference is written to the inverter reference percentage (*Reference Percentage RAM*). The *Reference Percentage* is calculated from *Torque Reference* [Nm], Parameter *Rated speed* **372** and Parameter *Rated Mech. Power* **376** in the active data set.



The EtherNet/IP™ objects *NetCtrl* (Class 0x29, Instance 1, Attribute 5) and *NetRef* (Class 0x2A, Instance 1, Attribute 4) **must** be set by explicit message from the PLC Program in order to set the inverter to the appropriate control mode.

NetCtrl sets Inverter Parameter *Local/Remote* **412** = 1 "Control via Statemachine" (RAM only)

NetRef sets Inverter Parameter *Ramp Setpoint* **434** = 2 "Line Setpoint" (RAM only)

This can be achieved by sending explicit messages to these objects individually, or setting both objects together by one explicit message, writing to *Class 4* (Assembly Object), *Instance 21* (Assembly 21), *Attribute 3* (Assembly Data).

For a description and example of how to send explicit messages from the PLC Program, please refer to Chapter Sending Explicit Messages from the PLC Program.

Ramp Setpoint 434	Function
1 - Internal setpoint	
2 - Line setpoint	
3 - Internal+Line setpoint	default

9.7.1.4 Output Assembly 24

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	Fault Reset	-	Run Forward
1	-	-	-	-	-	-	-	-
2	Speed Reference (low byte)							
3	Speed Reference (high byte)							
4	Process Reference (low byte)							
5	Process Reference (high byte)							

- unused, no function

Run Forward = 1 drive runs clockwise

Fault Reset = 0 → 1, acknowledge fault

Speed Reference = nnnn [RPM]

(written to inverter speed reference Line Setpoint, see **P.434**)

Speed Reference is written to the inverter reference frequency (*Reference Frequency RAM*, source no. 5,). The reference frequency is calculated from *Speed Reference* [RPM] and Parameter *No. of Pole Pairs* **373** in the active data set.

The data type of *Speed Reference* is INT. Setting *Speed Reference* to negative values causes the drive to run anticlockwise whenever *Run Forward* is set to 1.

Process Reference = nnnn [%]

Process Reference is written to the inverter reference percentage (*Reference Percentage RAM*,).



The EtherNet/IP™ objects *NetCtrl* (Class 0x29, Instance 1, Attribute 5) and *NetRef* (Class 0x2A, Instance 1, Attribute 4) **must** be set by explicit message from the PLC Program in order to set the inverter to the appropriate control mode.

NetCtrl sets Inverter Parameter *Local/Remote* **412** = 1 "Control via Statemachine" (RAM only)

NetRef sets Inverter Parameter *Ramp Setpoint* **434** = 2 "Line Setpoint" (RAM only)

This can be achieved by sending explicit messages to these objects individually, or setting both objects together by one explicit message, writing to *Class 4* (Assembly Object), *Instance 21* (Assembly 21), *Attribute 3* (Assembly Data).

For a description and example of how to send explicit messages from the PLC Program, please refer to Chapter Sending Explicit Messages from the PLC Program.

Ramp Setpoint 434	Function
1 - Internal setpoint	
2 - Line setpoint	
3 - Internal+Line setpoint	default

9.7.1.5 Input Assembly 70

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	Running1	-	Faulted
1	-	-	-	-	-	-	-	-
2	Speed Actual (low byte)							
3	Speed Actual (high byte)							

- unused, no function

Faulted = 1 fault detected

Running1 = 1 drive running

***Speed Actual* = nnnn [RPM]**

The data type of Speed Actual is INT. *Speed Actual* has negative values when the drive is running anticlockwise.

9.7.1.6 Input Assembly 71

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (Rev)	Running1 (Fwd)	Warning	Faulted
1	Drive State							
2	Speed Actual (low byte)							
3	Speed Actual (high byte)							

Faulted = 1 fault detected

Running1 (Fwd) = 1 drive running forward

Running2 (Rev) = 1 drive running reverse

Ready = 1 drive state Ready

Ctrl From Net = 1 inverter controlled by EtherNet/IP™

Ctrl From Net = 0 inverter controlled by local control

Ref From Net = 1 speed reference from EtherNet/IP™

Ref From Net = 0 speed reference from local control

Drive State

- 1 - Startup
- 2 - Not_Ready
- 3 - Ready
- 4 - Enabled
- 5 - Stopping
- 6 - Fault_Stop
- 7 - Faulted

Speed Actual = nnnn [RPM]

The data type of *Speed Actual* is INT. *Speed Actual* has negative values when the drive is running anticlockwise.

9.7.1.7 Input Assembly 72

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	Running1	-	Faulted
1	-	-	-	-	-	-	-	-
2	Speed Actual (low byte)							
3	Speed Actual (high byte)							
4	Torque Actual (low byte)							
5	Torque Actual (high byte)							

- unused, no function

Faulted = 1 fault detected

Running1 = 1 drive running

Speed Actual = nnnn [RPM]

The data type of *Speed Actual* is INT. *Speed Actual* has negative values when the drive is running anticlockwise.

Torque Actual = nnnn [Nm]

9.7.1.8 Input Assembly 74

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0						Running1		Faulted
1								
2	Speed Actual (low byte)							
3	Speed Actual (high byte)							
4	Process Actual (low byte)							
5	Process Actual (high byte)							

- unused, no function

Faulted = 1 fault detected

Running1 = 1 drive running

Speed Actual = nnnn [RPM]

The data type of *Speed Actual* is INT. *Speed Actual* has negative values when the drive is running anticlockwise.

Process Actual = nnnn [%]

9.7.2 Vendor Specific Assemblies 100/101

The vendor specific assemblies 100 and 101 allow the creation of assemblies with variable length and variable object mapping. The method is similar to PDO mapping with CANopen. A list of CANopen objects is available for mapping.



The functions connected to these objects are the same as with CANopen. For detailed information about available CANopen objects please refer to Chapter "CANopen objects".

Both assemblies (Output Assembly 100 and Input Assembly 101) support a maximum number of 12 objects with a maximum number of 24 bytes I/O data.

The selection of these vendor specific assemblies is performed in the same way as for CIP assemblies.



These assemblies are only settable if an IO-Connection is **not** established.

Output Assembly 100:

- Parameter Object Class 0x0F, Instance 15, Attribute 1 = **100**
- Parameter Object Class 0x0F, Instance 16, Attribute 1 = *Output Assembly length*

Input Assembly 101:

- Parameter Object Class 0x0F, Instance 1, Attribute 1 = **101**
- Parameter Object Class 0x0F, Instance 2, Attribute 1 = *Input Assembly length*

The assemblies can also be selected with the inverter configuration utility VPlus. The relevant parameters are located under Communication – CANopen / EtherNet/IP™:

- **P.1445** Input Assembly = **101**
- **P.1446** Input Assembly Len = *Input assembly length*
- **P.1447** Output Assembly = **100**
- **P.1448** Output Assembly Len = *Output assembly length*

9.7.2.1 Object Mapping - Vendor Specific Output Assembly 100**With VPlus / VTable**

When using the inverter utility VTable, Parameter **1429** Indices 1 to 24 offer a selection list where one of the listed objects can be chosen for mapping. Choosing "0 - no object mapped here" marks the end of the mapped objects.

See below for the object selection list.



The mapping **MUST** begin with Index 1. Unused Indices / Mapping Positions must be set to "- no object mapped here".

All mapping settings unequal to "- no object mapped here" following the first Index after "- no object mapped here" are ignored.

The mapping parameters are only settable if an IO-Connection is **not** established or the Connection is **not** set to Assembly 100/101.

Mapping example:

Index 1 - 0x6040 (2B) controlword

Index 2 - 0x6042 (2B) target velocity

Index 3 - no object mapped here → end of mapping, control word and target velocity are included in output assembly 100 and processed

Index 4 - 0x3003 (1B) Digital Out set values **IGNORED**

Index 5 **IGNORED**

...

Index 12 **IGNORED**



If objects are ignored, there will be **no error message** generated by VPlus. The ignored objects are automatically overwritten with "no object mapped here", however this is only visible after the next screen update.

Object Selection List

The values displayed in the selection list are:

- 0 "- no object mapped here",
- 43 "0x3003 (1B) Digital Out set values",
- 45 "0x3005 (2B) Boolean Demux",
- 46 "0x3006 (2B) Percentage set value",
- 53 "0x3111 (2B) Ref. value word1 (S.762)",
- 54 "0x3112 (2B) Ref. value word2 (S.763)",
- 55 "0x3121 (4B) Ref. value long1 (S.764)",
- 56 "0x3122 (4B) Ref. value long2 (S.765)",
- 57 "0x5F10_1 (2B) Numerator",
- 58 "0x5F10_2 (2B) Denominator",

- 80 "0x6040 (2B) controlword",
- 82 "0x6042 (2B) target velocity",
- 93 "0x6060 (1B) modes of operation",
- 105 "0x6071 (2B) Target torque",
- 109 "0x607A (4B) target position",
- 111 "0x6081 (4B) profile velocity",
- 112 "0x6083 (4B) profile acceleration",
- 113 "0x6084 (4B) profile deceleration",
- 114 "0x6085 (4B) quick stop deceleration",
- 128 "0x60FF (4B) target velocity";

0xnnnn:	CANopen object index
(xB):	number of bytes (object length)
Text:	CANopen object name

9.7.2.2 Object Mapping - Vendor Specific Input Assembly 101

With VPlus / VTable

When using the inverter utility VTable, Parameter **1430** Indices 1 to 24 offer a choice box where one of the listed objects can be chosen for mapping. Choosing "0 - no object mapped here" marks the end of the mapped objects.

See below for the object selection list.



The mapping **MUST** begin with Index 1. Unused Indices / Mapping Positions must be set to "- no object mapped here".

All mapping settings unequal to "- no object mapped here" following the first Index after "- no object mapped here" are ignored.

The mapping parameters are only settable if an IO-Connection is **not** established or the Connection is **not** set to Assembly 100/101.

Mapping example:

Index 1 - 0x6041 (2B) statusword

Index 2 - 0x6044 (2B) control effort

Index 3 - no object mapped here → end of mapping, status word and control effort are included in input assembly 101 and processed

Index 4 - 0x3001 (2B) Digital In actual values **IGNORED**

Index 5 **IGNORED**

...

Index 12 **IGNORED**



If objects are ignored, there will be **no error message** generated by VPlus . The ignored objects are automatically overwritten with "no object mapped here", however this is only visible after the next screen update.

Object Selection List

The values displayed in the selection list are:

- 0 "- no object mapped here",
- 41 "0x3001 (2B) Digital In actual values",
- 42 "0x3002 (2B) Digital Out actual values",
- 44 "0x3004 (2B) Boolean Mux",
- 47 "0x3007 (2B) Percentage actual value",
- 48 "0x3008 (2B) Percentage actual value2",
- 49 "0x3011 (2B) Act. value word1 (P.1415)",
- 50 "0x3012 (2B) Act. value word2 (P.1416)",

- 51 "0x3021 (4B) Act. value long1 (P.1417)",
- 52 "0x3022 (4B) Act. value long2 (P.1418)",
- 76 "0x5FF0 (1B) active motion block (ttr)",
- 77 "0x5FF1 (1B) motion block to resume (ttr)",
- 81 "0x6041 (2B) statusword",
- 83 "0x6043 (2B) velocity demand",
- 84 "0x6044 (2B) control effort",
- 94 "0x6061 (1B) modes of operation display",
- 95 "0x6064 (4B) position actual value",
- 100 "0x606C (4B) velocity actual value",
- 106 "0x6077 (2B) Torque actual value",
- 107 "0x6078 (2B) Current actual value",
- 108 "0x6079 (4B) DC link voltage",
- 126 "0x60F4 (4B) following error value";

0xnnnn:	CANopen object index
(xB):	number of bytes (object length)
Text:	CANopen object name

9.7.3 Handling of I/O Assemblies, Setup Device/PLC

All samples and screenshots in this chapter are taken from the Rockwell PLC.

CompactLogix L23E QBFC1B

with RSLogix 5000 (V20.01).



For the support/use of EDS files, RSLogix 5000 version V20 or higher is required.

STEP1 – Initial Inverter Setup

The inverter must be equipped with the module CM-EtherNet/IP. Connect the inverter to the network and assign the IP Address (refer to Chapter 6.2 "Network configuration for Ethernet/IP").

The required I/O Assemblies must first be selected with VPlus (see Chapter I/O Assemblies).

If vendor specific Assemblies 100/101 are selected, then check/select the mapped objects with VPlus-VTable (see Chapter Vendor Specific Assemblies 100/101).



From here on, the meaning of parameters is from EtherNet/IP™'s point of view. These EtherNet/IP™ parameters are accessible **via the network only**.

STEP2 – Register the Inverter's EDS File

- Connect the configuration PC to the PLC and start RSLogix 5000.
- Create a new project or select an existing project.
- Register the EDS file(s) for the used inverter(s). The necessary EDS files are created by the VPlus software tool (to do this a vcb-file of the inverter must be available or VPlus must be connected to the inverter). Please refer to Chapter How to create EDS Files.
- To register the new device, select "EDS Hardware Installation Tool" in the Tools menu.

The EDS Wizard is then started:

- Follow the instructions in the EDS Wizard.
- After successful registration, continue with the creation of the new Module:

STEP3 – Create the New Module

In the Controller Organizer window:

- right click on the "Ethernet" branch and select "New Module" in the menu list:

The "Select ModuleType" dialog box is displayed, with a list of all registered modules in the database:

- Enter the Catalog Number of the device in the "Enter Search Text for Module Type..." box.
The corresponding Catalog numbers are automatically displayed while you type.
- Whenever the required number is displayed, select it and then click the "Create" button.
The "New Module" dialog box is displayed:
- Enter the Module Name, the IP Address and then click on the "Change" button
This opens the "Module Definition" dialog box:
The Module Revision is automatically displayed (taken from the EDS File) and can be changed.
- Select the used Assemblies from the drop-down list box, and then set the Assembly Sizes and Tag Suffix, if necessary.

The size for the Vendor Specific Assemblies **100/101** can be selected (all others are fixed).

- Finally, check the Tag Suffix and change it, if necessary.
- Click the "OK" button to accept the settings or click "Cancel" to quit.

If changes were made a message box is displayed for confirmation.

The "Module Definition" dialog box is then closed and the "New Module" dialog box reactivated:

- Select the "Connection" tab for further settings:
- Click "Ok" to accept any changes or "Cancel" to quit.

9.8 Sending Explicit Messages from the PLC Program

This chapter describes how to send explicit messages from the RSLogix5000 PLC MainProgram. The following steps are required:

STEP1

- Register the Inverter EDS File and create the Inverter Module as described in Chapter 9.7.3.

STEP2

- Define the required messages (see Chapter 9.8.1 "How to create a MESSAGE Tag").

STEP3

- Add the appropriate code to the PLC MainProgram (see Chapter 9.8.2 "PLC Program Code Examples")

9.8.1 How to create a MESSAGE Tag

A Message is defined by creating a Controller Tag of Data Type *MESSAGE*.

This chapter describes the basic steps required for creating a Message Tag. For further details, please refer to the RSLogix 5000 Manual or Online Help.

The example below describes how to create the following Message:

Name: MsgCtrlWrite
Service Code: 0x10, Set Attribute Single
Class: 4, Assembly Object
Instance: 21, Output Assembly 21 (see Chapter 0)
Attribute: 3, Data value

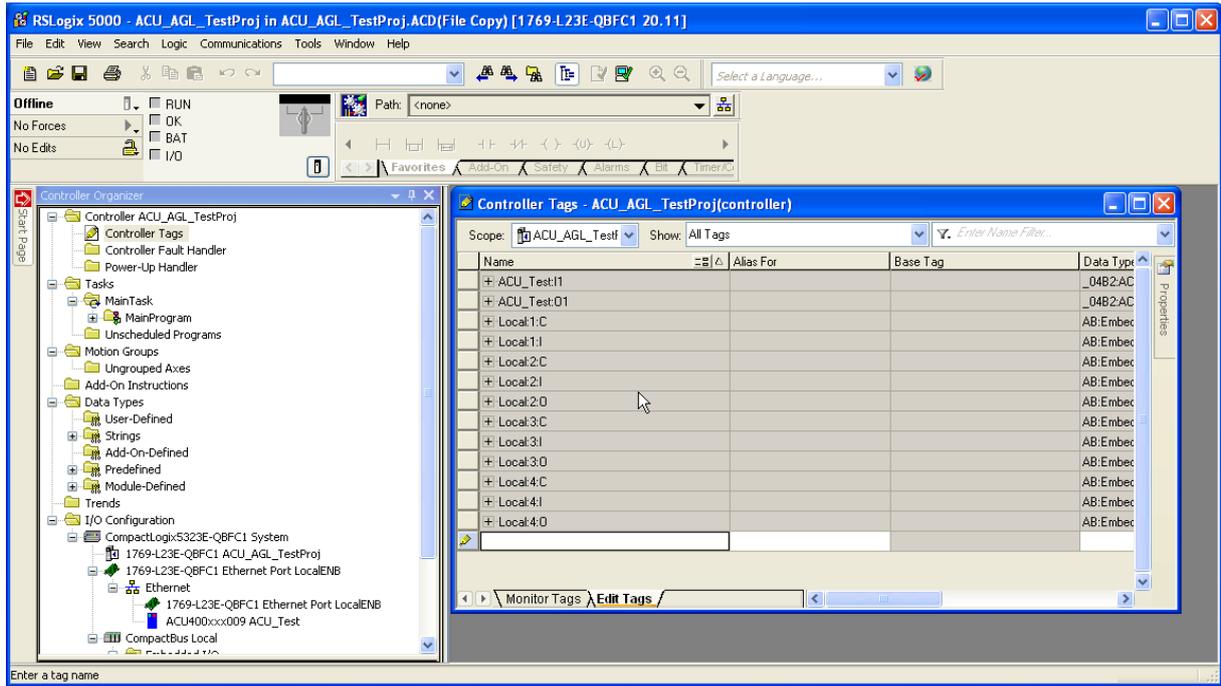
The following four data bytes are predefined for Assembly 21.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	Net Ref	Net Ctrl	-	-	Fault Reset	Run Reverse	Run Forward
1	-	-	-	-	-	-	-	-
2	Speed Reference (low byte)							
3	Speed Reference (high byte)							

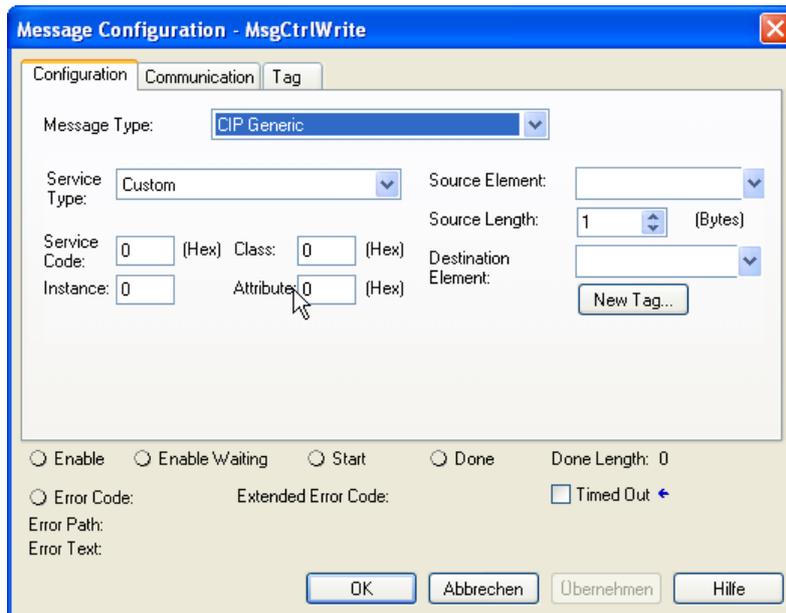
- unused, no function

This Message can be used to set the *NetCtrl* and *NetRef* objects when using Output Assembly 20, 22 or 24 (see Chapters 9.7.1.1, 9.7.1.3 and 9.7.1.4) as these Assemblies do not contain the *NetCtrl* and *NetRef* objects.

- Open the Controller Tags window, and click on the Edit Tags tab to enter the new tag:

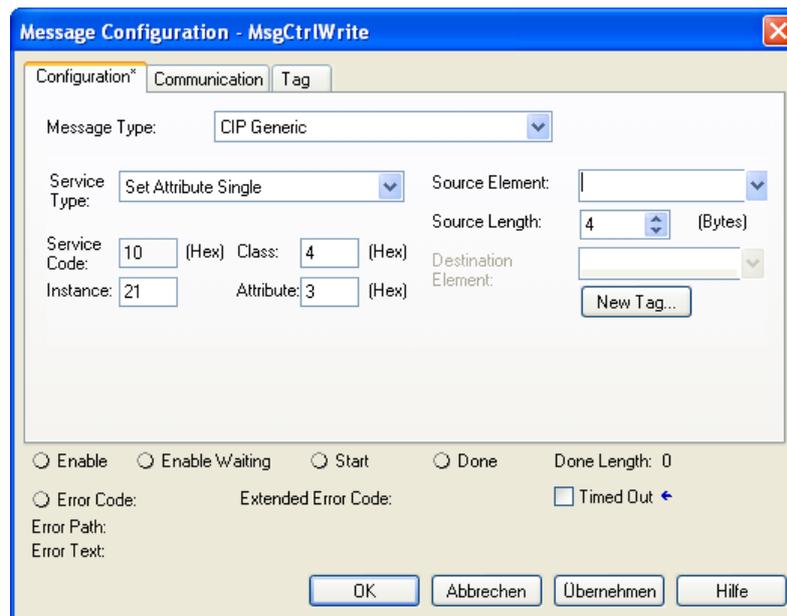


- Enter the Tag name "MsgCtrlWrite" in the data field provided and enter or select the Data Type "MESSAGE".
- Right click on the Tag Selection button to the left of the Tag Name and select Configure "MsgCtrlWrite" from the menu to display the Message Configuration dialog box:



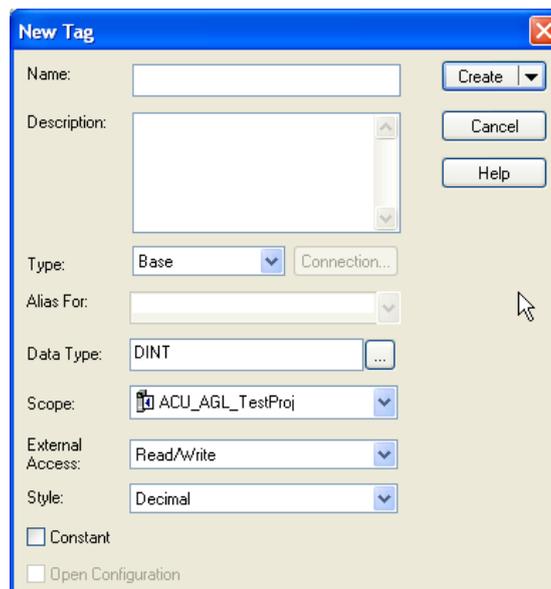
- Select Service Type "Set Attribute Single" from the pull-down menu. The Service Code (10) is displayed automatically.

- Set the Class to 4, Instance to 21, Attribute to 3 and the Source Length to 4.

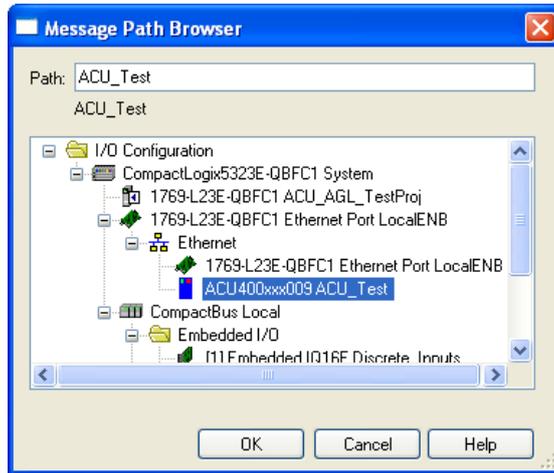


If not already defined, a new Tag must be defined for the Source Element:

- Click on the "New Tag" button to display the "New Tag" dialog box:



- Enter the following data:
 - Name: MsgDataCtrlWrite
 - Data Type: SINT[4]
- Then click on the "Create" button (or one of the options) to create the tag.
- Close the "New Tag" dialog box, if not done automatically, and select the newly created tag for the Source Element.
- Select the "Communication" tab and click on the "Browse" button to select the Message Path:



- Select the required device and Click "OK" to accept the selection and OK again in the "Message Configuration" dialog box.

The message is now created.

Other messages can be created in the same manner.

Once all messages are created, code can be added to the PLC Program to send the messages. This is described in the next chapter.

9.8.2 PLC Program Code Examples

Example 1: Setting *NetCtrl* and *NetRef* via Output Assembly 21.

This example uses the following Tags, created as described in the previous chapter:

MsgCtrlWrite

Tag of Data Type MESSAGE, defined as follows:



MsgDataCtrlWrite

Tag of Data Type SINT[4] – 4 Byte Buffer for the Message Data as defined by Output Assembly 21.

The following code sets the required data and sends the message:

```
// Set Assembly Data
MsgDataCtrlWrite[0]      := 16#60;    // NetRef = 1, NetCtrl = 1
MsgDataCtrlWrite[1]      := 0;        // Byte 1 not used
MsgDataCtrlWrite[2]      := 0;        // Speed Reference (low byte)
MsgDataCtrlWrite[3]      := 0;        // Speed Reference (low byte)
// Set Data Length
MsgCtrlWrite.REQ_LEN     := 4;        // length of Assembly 21
```

```
// Send Message
```

```
MSG(MsgCtrlWrite);
```

The following flags of the `MsgCtrlRead` message tag should then be monitored to evaluate the message state:

Flag	Name	Description
.EN	Enable	Message processing enabled
.DN	Done	Message sent successfully
.ER	Error	Message error The error code and extended error code are in the <code>.ERR</code> and <code>.EXERR</code> data fields of the <code>MsgCtrlRead</code> message structure.

For a full description of the `MSG` instruction, `MESSAGE` structure and other related topics, please refer to the RSLogix 5000 Manuals or Online Help

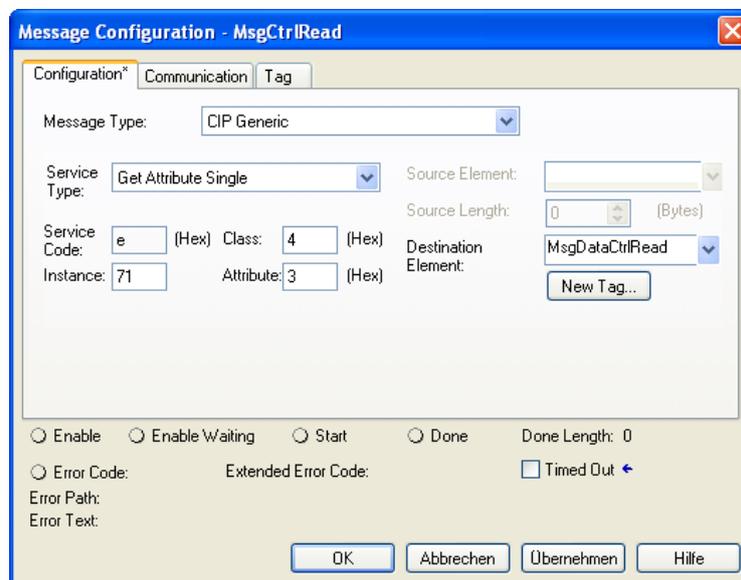
Example 2: Reading back `NetCtrl` and `NetRef` via Input Assembly 71.

This can be used for verification of the data sent as described in Example 1.

The following Tags are required:

MsgCtrlRead

Tag of Data Type `MESSAGE`, defined as follows:



MsgDataCtrlRead

Tag of Data Type `SINT[4]` 4 byte Buffer for the Message Data as defined by Input Assembly 71.

The following code sends the read request message:

```
// Read NetCtrl and NetRef via Assembly 71
// -----
// Send Message
MSG(MsgCtrlRead);
```

The following flags of the `MsgCtrlRead` message tag can then be monitored to evaluate the message state:

Flag	Name	Description
.EN	Enable	Message processing enabled
.DN	Done	Message sent successfully The data read is in tag <code>MsgDataCtrlRead</code>
.ER	Error	Message error The error code and extended error code are in the <code>.ERR</code> and <code>.EXERR</code> data fields of the <code>MsgCtrlRead</code> message structure.

For a full description of the *MSG* instruction, *MESSAGE* structure and other related topics, please refer to the RSLogix 5000 Manuals or Online Help.

9.9 Diagnosis of EtherNet/IP™ state

The EtherNet/IP™ State is displayed via the Inverter actual value Parameter *Node-State* **1290**.

10 CANopen Overview

CANopen® is used in a wide range of applications and is an especially favored 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 0xnxxx (16 bit) and sub-index 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**).

10.1.1 Communication Objects

The communication objects are located in the index range 0x1xxx. 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.

10.1.2 Application Objects

Application objects are divided in two groups. Index range 0x2000 to 0x5FFF is reserved for manufacturer-specific objects, and index range 0x6xxx is reserved for specific device profile objects. The specific device profile objects 0x6xxx are defined by CANopen® DS402 “drive and motion control”. They are used for controlling device functions (Start/Stop, speed, positioning functions).

10.1.3 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 9.9.3 “Manufacturer objects (0x2xxx)” 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 6.3.2.

10.1.3.1 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

10.1.3.2 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

NOTICE

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 9.9.3.1 "Handling of data sets/cyclic writing".

10.1.3.3 Error codes

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

The fault codes are listed in chapter 6.3.2 "SDO Error code table".

10.1.3.4 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 bytes per telegram are sent 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.

10.1.3.5 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 9.7.3.1)
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 →	0x600 + cs	Data						
	Server	Node-ID 0x70	00	00	00	00	00	00	00
Response	Server →	0x580 + cs	Data						
	Client	Node-ID 0x10	LSB	MSB

Segment Upload, last segment

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

10.1.3.6 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.

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
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 (Segmented) Transfer 1 = Expedited Transfer (see chapter 9.7.3.1)
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.

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
Response	Server →	0x580 +	cs Data						
	Client	Node-ID	0x20	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
Response	Server →	0x580 +	cs Data						
	Client	Node-ID	0x30	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
Response	Server →	0x580 +	cs Data						
	Client	Node-ID	0x10 oder 0x20	LSB

10.1.4 PDO Function

PDO (Process Data Objects) messages contain up to 8 bytes of process data. Using communication objects (communication/mapping parameters) the process data objects are mapped to Rx/Tx-PDOs. The frequency inverters support three RxPDOs (PLC → frequency inverter) and three TxPDOs (frequency inverter → PLC).

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

PDO-message:

Byte	0	1	2	3	4	5	6	7
	data							

The number of bytes is 1 ... 8 and depends on the mapped objects. The bytes are arranged in the 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

10.1.5 Emergency Function

If there is a communication error or an error in the frequency inverter, the frequency inverter will send an error message. The error message contains the relevant error information. Once the error is acknowledged (error reset), an error message is sent, with the data bytes reset 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 "0x1014/0 COB-ID Emergency Message".



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

10.1.6 SYNC (synchronous transmission)

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 6.3 "OS Synchronization".

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

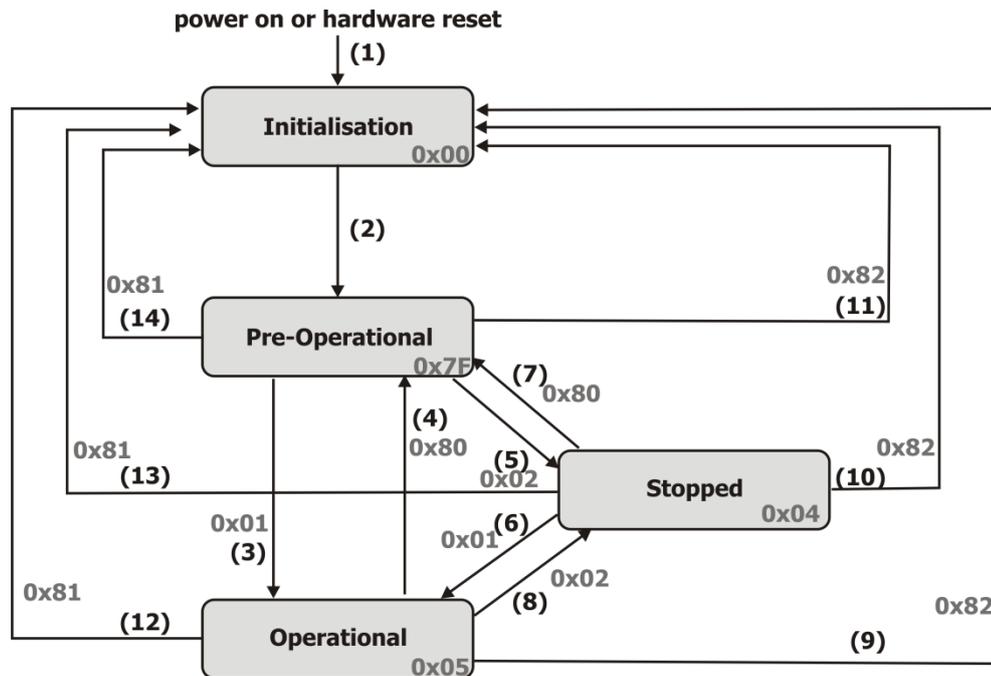
10.1.7 NMT Functions

The NMT (= Network Management) functions describe the NMT Statemachine and NMT error saving 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**.

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 9.8.9 "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.

10.1.7.1 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 10.1.8 "Guarding".

10.1.7.2 NMT commands

Identifier	Byte 0	Byte 1
	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

10.1.8 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:

Identifier	Byte 0							
	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

10.1.9 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 structure:

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

10.2 Special CANopen objects

CANopen® uses for Communication several objects that are not used with Ethernet I/P®. These are described in this chapter.



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

Orange color	= Rd Only object
Green color	= Read and Write object
Blue color	= Write only object

Abbreviations used

Access: Access type
r/w: R/W
ro: Rd Only
wo: Write only

Map: Mapping
Def.-Val: Default value

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 Object name*.



The objects which can be used for either Ethernet I/P® or CANopen® are described in chapter 9.9 "CANopen objects" to prevent double descriptions in this manual.

10.2.1 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

10.2.2 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.

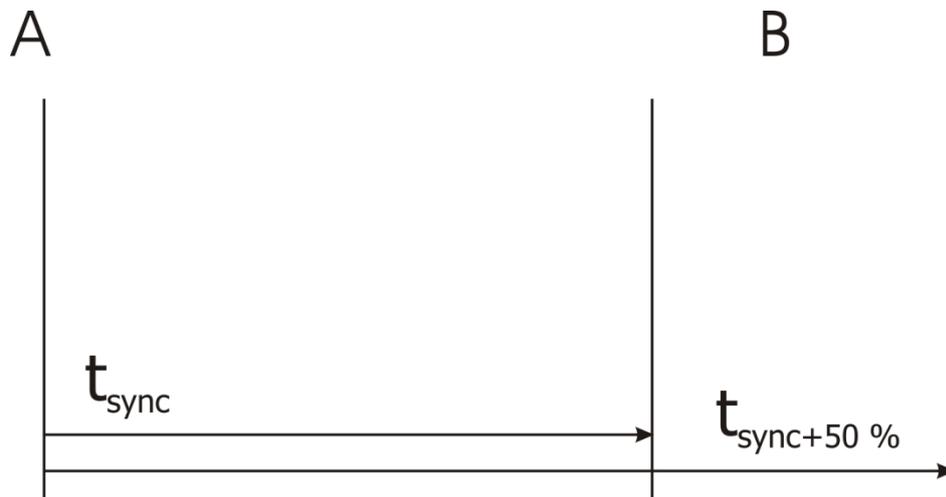
The synchronization of the inverter to an external clock has to be met under the condition, that at least one RxPDO or TxPDO is defined as synchronous object and is activated. The definitions of the TxPDO / RxPDO objects can be changed via objects [0x1400](#) / [0x1800](#).



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.



10.2.3 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 microseconds.

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*.

10.2.4 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 *guard 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](#).

10.2.5 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.

10.2.6 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.

10.2.7 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

10.2.8 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.

10.2.9 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

10.2.10 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

10.2.11 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

10.2.12 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 $300 * 100 \text{ us} = 30 \text{ ms}$.



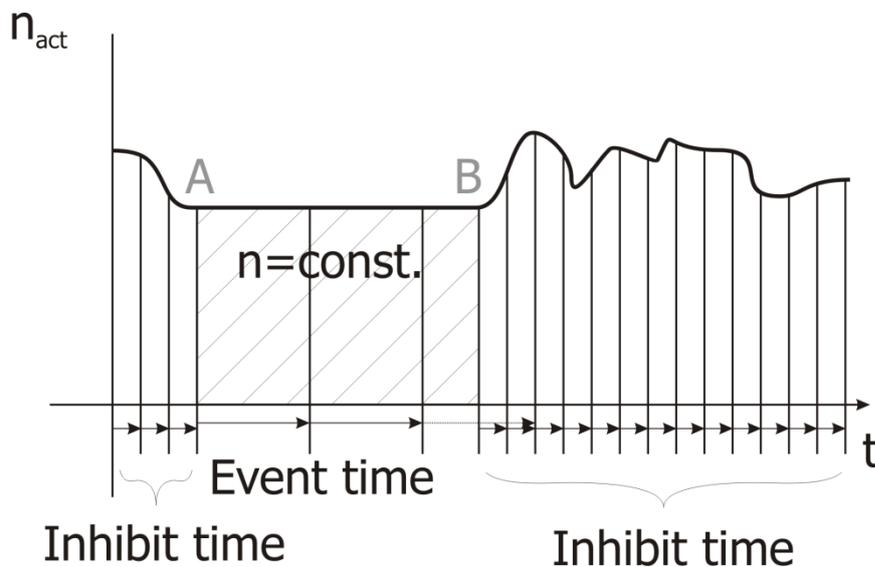
The device internal time resolution for the inhibit time is in milliseconds, the last digit is always converted to "0". An inhibit time value = 37 is truncated to 30 [$3.7 \text{ ms} \rightarrow 3 \text{ ms}$]. Values less than 10 are interpreted as 0.

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 & Inhibit 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.

10.2.13 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. ACU 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:

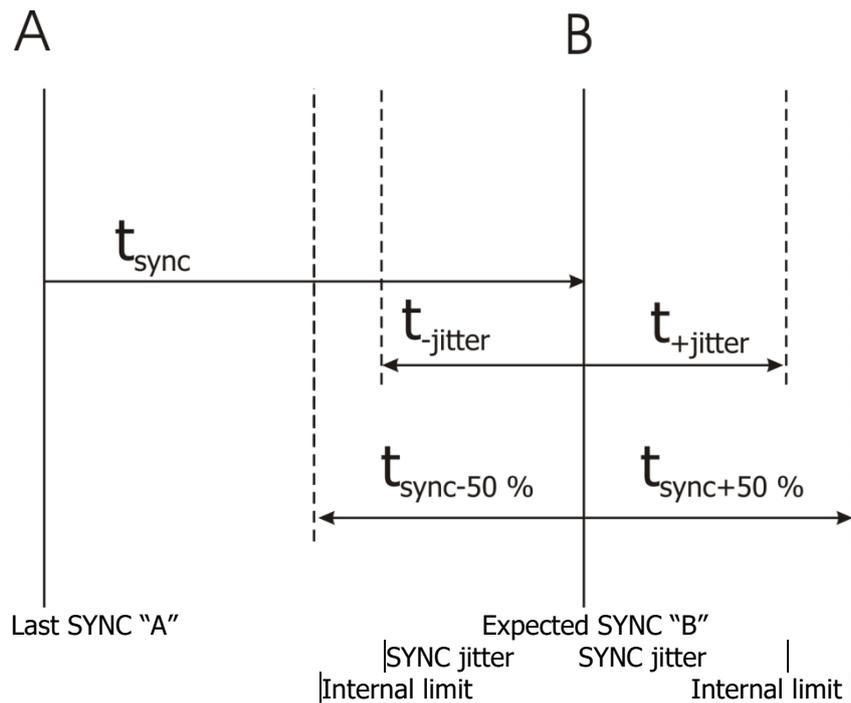
0x1006/0 communication cycle period +/- **0x3000/0 SYNC Jitter**

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).



10.3 CANopen objects



This Chapter "CANopen Objects" and the following Chapters "Motion Control Interface (MCI)" and "Control" are only relevant when using EtherNet/IP™ manufacturer specific assemblies 100/101.



Some parameters available to the user are not described in the following subchapters. They can be used to customize specific functions in application. Please contact the manufacturer for details.

10.3.1 Objects tabular overview

The objects are listed in the following tables. The following definitions apply:

Access type	
Rd Only	The PLC can only read data from the frequency inverter.
R/W	The PLC is granted unlimited access (reading and writing) to the frequency inverter 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 exchange of PDO. Only SDO can be used.		
Tx	This object can be transmitted from the frequency inverter in a TxPDO.		
Rx	This object can be transmitted to the frequency inverter in a RxPDO.		



"Highest sub index supported" shows the highest sub-index supported by the object.

10.3.1.1 Communication objects



The communication objects listed in the table below do not support PDO mapping.

Index	SubIndex	Name	SDO Acc.	Data type
0x1000	0	Device type	Rd Only	Unsigned32
0x1001	0	Error register	Rd Only	Unsigned8
0x1008	0	Manufacturer device name	Rd Only	Visible string
0x1009	0	Manufacturer hardware version	Rd Only	Visible string
0x100A	0	Manufacturer software version	Rd Only	Visible string
0x1010		Store parameters		
	0	Highest Sub-index supported	Rd Only	Unsigned8
	1	Save all parameters	R/W	Unsigned32
	2	Save communication parameters	R/W	Unsigned32
	3	Save application parameters	R/W	Unsigned32
0x1011		Restore default parameters		
	0	Highest Sub-index supported	Rd Only	Unsigned8
	1	Restore all default parameters	R/W	Unsigned32
	2	Restore communication default parameters	R/W	Unsigned32
	3	Restore application default parameters	R/W	Unsigned32
0x1018		Identity object		
	0	Highest Sub-index supported	Rd Only	Unsigned8
	1	Vendor ID	Rd Only	Unsigned32
	2	Product code	Rd Only	Unsigned32
	3	Revision number	Rd Only	Unsigned32
	4	Serial number	Rd Only	Unsigned32
0x1600		RxPDO1 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
	8	8. mapped obj.	R/W	Unsigned32
0x1601		RxPDO2 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
	8	8. mapped obj.	R/W	Unsigned32

Index	SubIndex	Name	SDO Acc.	Data type
0x1602		RxPDO3 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
8	8. mapped obj.	R/W	Unsigned32	
0x1A00		TxPDO1 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
8	8. mapped obj.	R/W	Unsigned32	
0x1A01		TxPDO2 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
8	8. mapped obj.	R/W	Unsigned32	
0x1A02		TxPDO3 mapping parameter		
	0	No. of mapped objects	R/W	Unsigned8
	1	1. mapped obj.	R/W	Unsigned32
	2	2. mapped obj.	R/W	Unsigned32
	3	3. mapped obj.	R/W	Unsigned32
	4	4. mapped obj.	R/W	Unsigned32
	5	5. mapped obj.	R/W	Unsigned32
	6	6. mapped obj.	R/W	Unsigned32
	7	7. mapped obj.	R/W	Unsigned32
8	8. mapped obj.	R/W	Unsigned32	

10.3.1.2 Manufacturer objects

Index	SubIndex	Name	SDO Acc.	Data type	PDO-mapping	Default	Min...Max
0x2nnn	0, 1, ... 9	Manufacturer specific Direct access to inverter parameters R/W access by SDO transfer only					
		Please refer to chapter	9.9.3.1 "Handling of data sets/cyclic writing of the parameters"				
0x3001	0	Digital In actual values	Rd Only	Unsigned16	Tx	-	-
0x3002	0	Digital Out actual values	Rd Only	Unsigned16	Tx	-	-
0x3003	0	Digital Out set values	R/W	Unsigned16	Rx	0	0...0x1F
0x3004	0	Boolean Mux	Rd Only	Unsigned16	Tx	-	-
0x3005	0	Boolean Demux	R/W	Unsigned16	Rx	0	0...0xFFFF
0x3006	0	Percentage set value	R/W	Unsigned16	Rx	0	0x8AD0... 0x7530
0x3007	0	Percentage actual value 1	Rd Only	Unsigned16	Tx	-	-

Index	SubIndex	Name	SDO Acc.	Data type	PDO-mapping	Default	Min...Max
0x3008	0	Percentage actual value 2	Rd Only	Unsigned16	Tx	-	-
0x3011	0	Act. value Word 1	Rd Only	Unsigned16	Tx	-	-
0x3012	0	Act. value Word 2	Rd Only	Unsigned16	Tx	-	-
0x3021	0	Act. value Long 1	Rd Only	Unsigned32	Tx	-	-
0x3022	0	Act. value Long 2	Rd Only	Unsigned32	Tx	-	-
0x3111	0	Ref. value Word 1	R/W	Unsigned16	Rx	0	0...0xFFFF
0x3112	0	Ref. value Word 2	R/W	Unsigned16	Rx	0	0...0xFFFF
0x3121	0	Ref. value Long 1	R/W	Unsigned32	Rx	0	0... 0xFFFF.FFFF
0x3122	0	Ref. value Long 2	R/W	Unsigned32	Rx	0	0... 0xFFFF.FFFF
0x5F10		Gear factor ⁹⁾					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1123	1	Numerator	R/W	Integer16	Rx	1	1...0x7FFF
P.1124	2	Denominator	R/W	Unsigned16	Rx	1	1...0xFFFF
P.1142	3	Resync on Change	R/W	Integer16	No	1	0...1
0x5F11		Phasing 1 ⁹⁾					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1125 DS1	1	Offset	R/W	Integer32	No	0x0001.0000	0x8000.0001 ... 0x7FFF.FFFF
P.1126 DS1	2	Speed	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
P.1127 DS1	3	Acceleration	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
0x5F12		Phasing 2 ⁹⁾					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1125 DS2	1	Offset	R/W	Integer32	No	0x0001.0000	0x8000.0001 ... 0x7FFF.FFFF
P.1126 DS2	2	Speed	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
P.1127 DS2	3	Acceleration	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
0x5F13		Phasing 3 ⁹⁾					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1125 DS3	1	Offset	R/W	Integer32	No	0x0001.0000	0x8000.0001 ... 0x7FFF.FFFF
P.1126 DS3	2	Speed	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
P.1127 DS3	3	Acceleration	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
0x5F14		Phasing 4 ⁹⁾					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1125 DS4	1	Offset	R/W	Integer32	No	0x0001.0000	0x8000.0001 ... 0x7FFF.FFFF
P.1126 DS4	2	Speed	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
P.1127 DS4	3	Acceleration	R/W	Unsigned32	No	0x0005.0000	1... 0x7FFF.FFFF
0x5F15	0	In Gear Threshold	R/W	Unsigned32	No	0	0...0x7FFF.FFFF
P.1168							
0x5F16	0	In Gear Time	R/W	Unsigned16	No	10	1...0xFFFF
P.1169							

Index	SubIndex	Name	SDO Acc.	Data type	PDO-mapping	Default	Min...Max
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1104	1	Time Constant [ms]	R/W	Integer32	No	10,00	1,00...300,00
P.1118	2	Limitation	R/W	Unsigned32	No	327680	0... 0x7FFF.FFFF
0x5F18 P.1284	0	Master Synchronization Offset ^{g)}	R/W	Integer32	No	0	0x8000.0001 ... 0x7FFF.FFFF
0x5FF0	0	Active motion block ^{t)}	Rd Only	Unsigned8	Tx	-	-
0x5FF1	0	Motion block to resume ^{t)}	Rd 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.

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.

10.3.1.3 Device profile objects

Index	SubIndex	Designation	SDO Access	Data type	PDO-mapping	Default setting	Min...Max
0x6007 P.388	0	Abort connection option code	R/W	Integer16	No	1	-2...3
0x603F	0	Error code	Rd Only	Unsigned16	No	-	-
0x6040 P.410	0	Control word	R/W	Unsigned16	Rx	-	-
0x6041 P.411	0	Status word	Read/only	Unsigned16	Tx	-	-
0x6042	0	vl target velocity ^{v)}	R/W	Integer16	Rx	0	- 32768...32767
0x6043	0	vl velocity demand	Rd Only	Integer16	Tx	-	-
0x6044	0	vl velocity actual value	Rd Only	Integer16	Tx	-	-
0x6046		vl velocity min max amount					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.418	1	vl velocity min amount	R/W	Unsigned32	No	0	0...32767
P.419	2	vl velocity max amount	R/W	Unsigned32	No	32767	0...32767
0x6048		vl velocity acceleration ^{v)}				-	-
	0	Highest sub-index supported	Rd Only	Unsigned8	No		
P.420	1	Delta speed	R/W	Unsigned32	No	150	1...32767
P.422	2	Delta time	R/W	Unsigned16	No	1	1...65535
0x6049		vl velocity deceleration ^{v)}					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.421	1	Delta speed	R/W	Unsigned32	No	150	1...32767
P.423	2	Delta time	R/W	Unsigned16	No	1	1...65535
0x604A		vl velocity quick stop ^{v)}					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.424	1	Delta speed	R/W	Unsigned32	No	150	1...32767
P.425	2	Delta time	R/W	Unsigned16	No	1	1...65535

Index	SubIndex	Designation	SDO Access	Data type	PDO-mapping	Default setting	Min...Max
0x6060	0	Modes of operation	Write only	Integer8	Rx	2	-3...9
0x6061	0	Modes of operation display	Rd Only	Integer8	Tx	2	-
0x6064 P.1108	0	Position actual value	Rd Only	Integer32	Tx	-	0x8000.0001 ... 0x7FFF.FFFF
0x6065 P.1105	0	Following error window	R/W	Unsigned32	No	0xFFFF.F FFF	0... 0xFFFF.FFFF
0x6066 P.1119	0	Following error time out	R/W	Unsigned16	No	10	0...65535
0x6067 P.1165	0	Position window	R/W	Unsigned32	No	0xFFFF.F FFF	0... 0xFFFF.FFFF
0x6068 P.1166	0	Position window time	R/W	Unsigned16	No	10	0...65535
0x606C	0	Velocity Actual value ^{u)}	Read	Integer32	Tx		
0x606D P.1276	0	Velocity Window ^{u)}	R/W	Unsigned16	No	1000	0...65535
0x606E P.1277	0	Velocity Window Time ^{u)}	R/W	Unsigned16	No	0	0...65535
0x606F P.1278	0	Velocity Threshold ^{u)}	R/W	Unsigned16	No	100	0...65535
0x6070 P.1279	0	Velocity Threshold Time ^{u)}	R/W	Unsigned16	No	0	0...65535
0x6071	0	Target torque	R/W	Integer16	Rx		
0x6077 P.224	0	Torque actual value	Rd Only	Integer16	Tx		
0x6078 P.214	0	Current actual value	Rd Only	Integer16	Tx		
0x6079 P.222	0	DC link circuit voltage	Rd Only	Integer32	Tx		
0x607A P.1202	0	Target position ^{p)}	R/W	Integer32	Rx	0	0x8000.0001 ... 0x7FFF.FFFF
0x607C P.1131	0	Home offset ^{h)}	R/W	Integer32	No	0	0x8000.0001 ... 0x7FFF.FFFF
0x6081	0	Profile velocity ^{p)} ^{u)}	R/W	Unsigned32	Rx	0x0005.0 000	1... 0x7FFF.FFFF
0x6083	0	Profile acceleration ^{p)} ^{u)}	R/W	Unsigned32	Rx	0x0005.0 000	1... 0x7FFF.FFFF
0x6084	0	Profile deceleration ^{p)} ^{u)}	R/W	Unsigned32	Rx	0x0005.0 000	1... 0x7FFF.FFFF
0x6085 P.1179	0	Quick stop deceleration ^{h)} ^{p)} ^{t)} ^{u)}	R/W	Unsigned32	No	0x000A.0 000	1... 0x7FFF.FFFF
0x6086	0	Motion profile type ^{u)}	R/W	Integer16	No	3	0...3
0x6091		Gear ratio					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1116	1	Motor revolutions	R/W	Unsigned32	No	1	1...65535
P.1117	2	(Driving) Shaft revolutions	R/W	Unsigned32	No	1	1...65535
0x6092		Feed constant					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1115	1	Feed	R/W	Unsigned32	No	0x0001.0 000	1... 0x7FFF.FFFF
	2	(Driving) Shaft revolutions	R/W	Unsigned32	No	1	1
0x6098 P.1130	0	Homing method ^{h)}	R/W	Integer8	No	0	0...35

Index	SubIndex	Designation	SDO Access	Data type	PDO-mapping	Default setting	Min...Max
0x6099		Homing speeds ^{h)} ^{l)}					
	0	Highest sub-index supported	Rd Only	Unsigned8	No	-	-
P.1132	1	Speed during search for switch	R/W	Unsigned32	No	0x0005.000	1... 0x7FFF.FFFF
P.1133	2	Speed during search for zero	R/W	Unsigned32	No	0x0002.000	1... 0x7FFF.FFFF
0x609A P.1134	0	Homing acceleration ^{h)}	R/W	Unsigned32	No	0x0005.000	1... 0x7FFF.FFFF
0x60F4 P.1109	0	Following error actual value	Rd Only	Integer32	Tx		
0x60F8 P.1275	0	Max Slippage ^{u)}	R/W	Integer32	No	0	
0x60FF	0	Target velocity ^{u)}	R/W	Integer32	Rx		
0x6502	0	Supported drive modes	Rd Only	Unsigned32	No	0x0202	0x0202

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.

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.

The Modes "*Homing*", "*Profile Position*", "*Profile Velocity*", "*Table travel record*" and "*Move away from Limit Switch*" require a configuration capable of Positioning. Check chapter Motion control configurations" for details.



The representations of CANopen® objects and parameters may be different (refer to relevant object description).

NOTICE

Unexpected behavior!

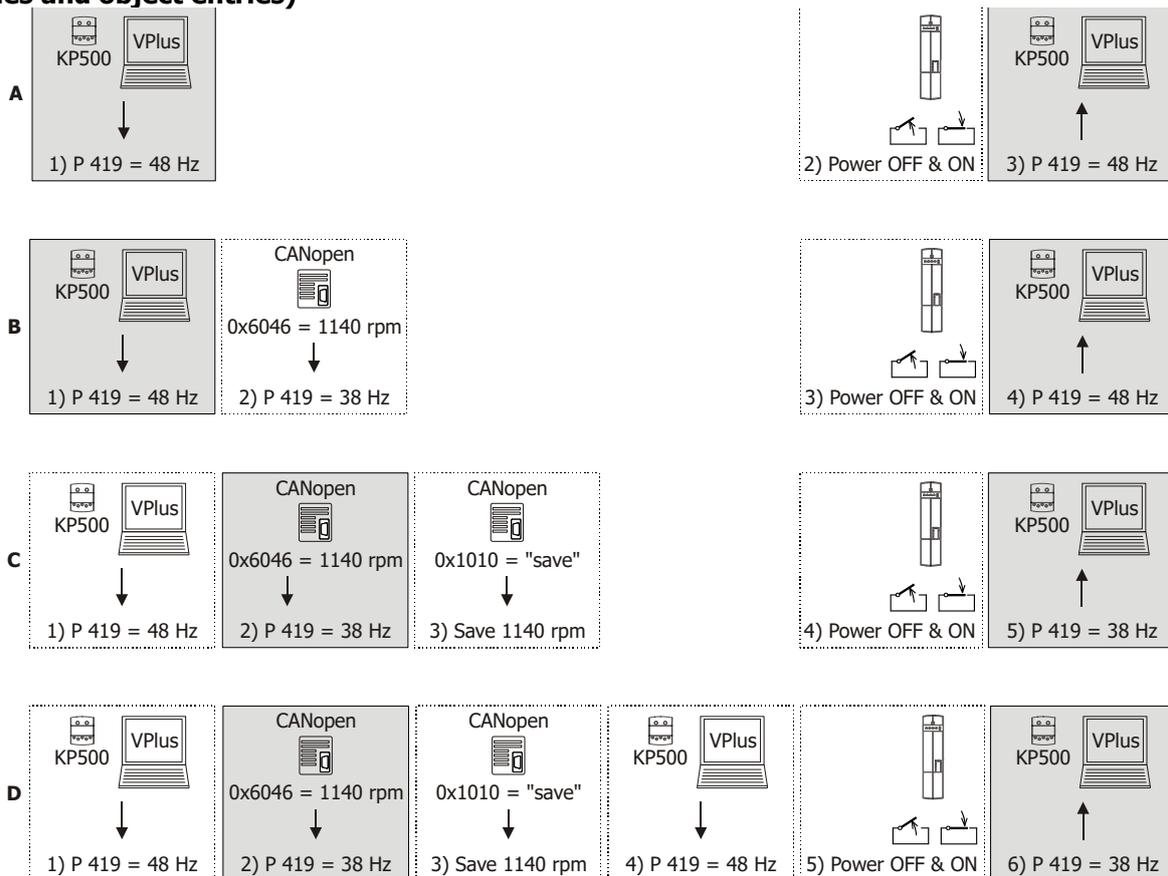
Some of the above listed CANopen® objects have corresponding inverter parameters. These objects are handled in a special way.

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 or KP500, the next power-on cycle overwrites the value set by VPlus with the value stored by the "save" command.

For more information on VPlus, see chapter 17.

Effect of the "save" command (Object [0x1010](#)) (Example of sequence of parameter entries and object entries)



Sequence

- 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).

NOTICE
Implausible results!

For some frequency parameters, which are calculated from CANopen® DS402 objects, it is necessary to enter the number of pole pairs, e. g. for calculation of the deceleration or acceleration parameters. These calculations use the number of pole pairs from Dataset 1. If the number of pole pairs in the datasets is different, the result of the calculation may be implausible to the user.

- For this reason, we recommend writing the frequency inverter parameters via the SDO channel with objects [0x2nnn](#) (manufacturer) and **not using** the CANopen® DS402 objects. In this way, inconsistencies are avoided.

CANopen® DS402 objects with corresponding frequency inverter parameters are indicated in this manual.

10.3.2 Communication Objects (0x1nnn)

Communication objects *0x1nnn* contain all parameters for CANopen communication.



To facilitate the overview, the objects are summarized by a table in each paragraph. This table is marked additionally by color.

Orange color	= Rd Only object
Green color	= Read and Write object
Blue color	= Write only object

Abbreviations used

Access: Access type
 r/w: R/W
 ro: Rd Only
 wo: Write only

Map: Mapping
 Def.-Val: Default value



The headings of the following chapters are displayed in the format *Index/Subindex Object name*.

10.3.2.1 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 upon network startup. The information about the device type and functionality (type) is defined by the CANopen® DS402 standard.

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

The standard device profile "*Drives and Motion Control*" used by the frequency inverter is shown as device profile number **402**. The other 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

10.3.2.2 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 frequency inverter errors. Status "no error" (0x1001/0 = 0) or "Error" (0x1001/0 ≠ 0) is displayed.

Detailed information on the device error can be read via VPlus, Parameter *Current error* **259** (see Chapter 18.4 "Error messages").

In case of an error, the PLC can evaluate detailed information via the Emergency Message (see Chapter "" and 9.9.5.2 "0x603F/0 Error code").

Object 0x1001/0	
Bit	
0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific error
6	Reserved
7	Manufacturer specific error

10.3.2.3 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"

10.3.2.4 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: "ACU400 512 344"

10.3.2.5 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: 8.0.5

10.3.2.6 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 three sub-indexes with various functions.

Writing of "save" in 0x1010/3 saves all application parameters ([0x6nnn](#)) in the non-volatile memory.

Specification for writing the "save" instruction

LSB			MSB
"s"	"a"	"v"	"e"
0x73	0x61	0x76	0x65



Writing of values other than "save" will result in cancellation of SDO. The store command is **not** processed.

10.3.2.7 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, you can reset parameters/objects to the default values. This object supports three sub-indexes with different functions.

Writing of "load" in 0x1011/3 restores all application parameters ([0x6nnn](#)).

Specification for writing the "load" instruction

LSB		MSB	
"l"	"o"	"a"	"d"
0x6C	0x6F	0x61	0x64



Writing of values other than "load" will result in cancellation of SDO. The restore defaults command is **not** processed.

10.3.2.8 0x1018/n Identity Object

The object *identity* provides information about the device manufacturer and the device.

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

"Vendor ID" "0xD5" refers to manufacturer **BONFIGLIOLI VECTRON MDS GmbH**.

This "Vendor ID" is assigned by the CiA (CAN in Automation) CANopen users' organization.

Product code:

shows the type ID of the frequency inverter.

Revision number:

shows the revision level of the CANopen® system of the frequency inverter.

Serial number:

shows the serial number of the frequency inverter.

10.3.2.9 0x1600/n, 0x1601/n, 0x1602/n, RxPDO Mapping Parameter

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

RxPDO Mapping parameters:

0x1600/n RxPDO1

0x1601/n RxPDO2

0x1602/n RxPDO3

0x1600/0 = 0 = no object mapped

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

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

Refer to chapter 9.9.1 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 v/target velocity	0x00000000
RxPDO2	0x1601/0	0x1601/1...8		
	1	No mapping		
RxPDO3	0x1602/0	0x1602/1...8		
	1	No mapping		

10.3.2.10 0x1A00/n, 0x1A01/n, 0x1A02/n, TxPDO Mapping Parameter

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

Mapping entry:

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

Examples:

Mapping of [0x6041/0 Statusword](#) (unsigned16) to "1st mapped obj." in TxPDO1:

0x1A00/1 = 0x60410010

Mapping of [0x6064/0 Position actual value](#) (integer32) to "2nd mapped obj." in TxPDO1:

0x1A00/2 = 0x60640020

Default mapping

TxPDO1	0x1A00/0	0x1A00/1	0x1A00/2	0x1A00/3...8
	2	0x6041 Status word	0x6044 v/velocity actual value	0x00000000

TxPDO2	0x1A01/0	0x1A01/1...8
	1	No mapping
TxPDO3	0x1A02/0	0x1A02/1...8
	1	No mapping



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

10.3.3 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)



Mapping of numerical data is always an integer or long data type. Values with decimal places will be written without decimal point (e.g. value 17.35 will be transmitted as 1735).

10.3.3.1 Handling of data sets/cyclic writing of the parameters

The parameter values are accessed based on the parameter number and the required dataset. 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 will be signaled.

NOTICE

Risk of component damage!

If a maximum number of write cycles for the EEPROM is exceeded, it will be destroyed. The values are entered automatically in the EEPROM of the controller. However, only a limited number of write cycles is permissible for the EEPROM (approx. 1 million cycles).

- If cyclic writing of data is required, use the RAM.

In the RAM, the data is not protected against loss of power. Once power supply is disrupted, the data must be written again.

This mechanism is activated by the target data set being increased by five in the specification of the data set.

Writing on virtual dataset in 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

10.3.3.2 Handling of index parameters/cyclic writing

NOTICE

Risk of component damage!

If a maximum number of write cycles for the EEPROM is exceeded, it will be destroyed. The values are entered automatically in the EEPROM of the controller. However, only a limited number of write cycles is permissible for the EEPROM (approx. 1 million cycles).

- If cyclic writing of data is required, use the RAM.

Index parameters are used for various frequency inverter functions. Here, 16 or 32 indexes are used instead of the 4 data sets. For each function, the individual indexes are addressed separately via an index access parameter. Via the indexing parameter, you can select if the data is to be written to EEPROM or RAM.

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 <i>Mux Input</i>	0 ¹ ; 1...16	17 ¹ ; 18...33	1250 Write 1251 Read
CANopen Multiplexer	1422 <i>CANopen Mux Input</i>	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.

In the RAM, the data is not protected against loss of power. Once the power supply is interrupted, the data must be written again.

This mechanism is activated by the target data set being increased by five in the specification of the data set.

Example Writing an index parameter

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

Writing Parameter *Target Position / Distance* **1202** (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



If various parameters of an index are to be edited, it will be sufficient to set index access via parameter **1200** once at the beginning.

Example Reading an index parameter

In order to read an index parameter, you will have to set the index parameter to the relevant index first, then you can read the parameter.

Reading Parameter *Target Position / Distance* **1202** (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



If various parameters in an index are to be edited, it will be sufficient to set index access via parameter **1201** once at the beginning.

10.3.4 Manufacturer objects (0x3000 ... 0x5FF1)

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

10.3.4.1 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* shows – like parameter *Digital inputs* **250** – the current state of the digital inputs and multifunction input **1** (when parameter *Operation mode* **452** is set to "3 – Digital input").

10.3.4.2 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* shows – like parameter *Digital outputs* **254** – the current state of the digital outputs and multifunction output **1** (when parameter *Operation mode* **550** is set to "1 – Digital"). The selection of digital outputs depends on the extension modules installed as an option.

10.3.4.3 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, five digital parameter sources are available requiring assignment of digital sources.

Object 0x3003			
Bit	Source number	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 values of object 0x3003 range from 0 to 31.

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

Digital outputs use these sources as operation modes 90 ... 94 with source names Obj 0x3003 DigOut 1 ... 5 and, inverted, as operation modes 190 ... 194 with source names inv. Obj 0x3003 DigOut 1 ... 5 (e.g. parameter *Op. Mode Digital In-/Output 1* **530**). These object bits can be mapped to the output as required.

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 ...

Sources 810...814 with source names Obj 0x3003 DigOut 1 ... 5 can be selected directly via the selection list for parameters. This can be used, for example, for direct setting of Boolean inputs.

10.3.4.4 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 packed Boolean values can be read. Each bit in 16-bit object 0x3004 shows the actual value of the assigned Boolean source.

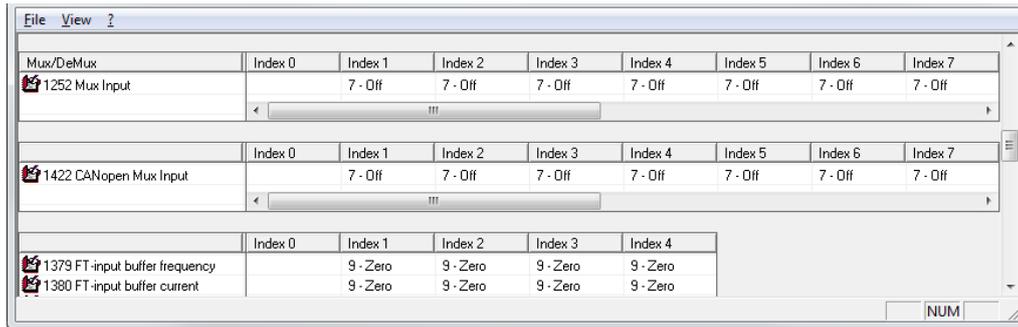


Bit numbers 0 ... 15 correspond to index numbers 1 ... 16!

The sources of the 16 bits can be selected from a choice list of index parameter *CANopen Mux Input* **1422**. Parameters **1420** and **1421** are write and read parameters which must be set before writing/reading of parameter **1422**.

By using VTable this process is easier and laid out more clearly.

For writing and reading index parameters refer to chapter 9.9.3.2 "Handling of index parameters/cyclic writing".



Default value is 7 - Off.

10.3.4.5 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 packed Boolean values can be written. These values are available as sources and can be selected via a choice list for parameters as objects 832...847 with source names Obj 0x3005 Demux Out 1...16.

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

10.3.4.6 0x3006/0 Percentage set value

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

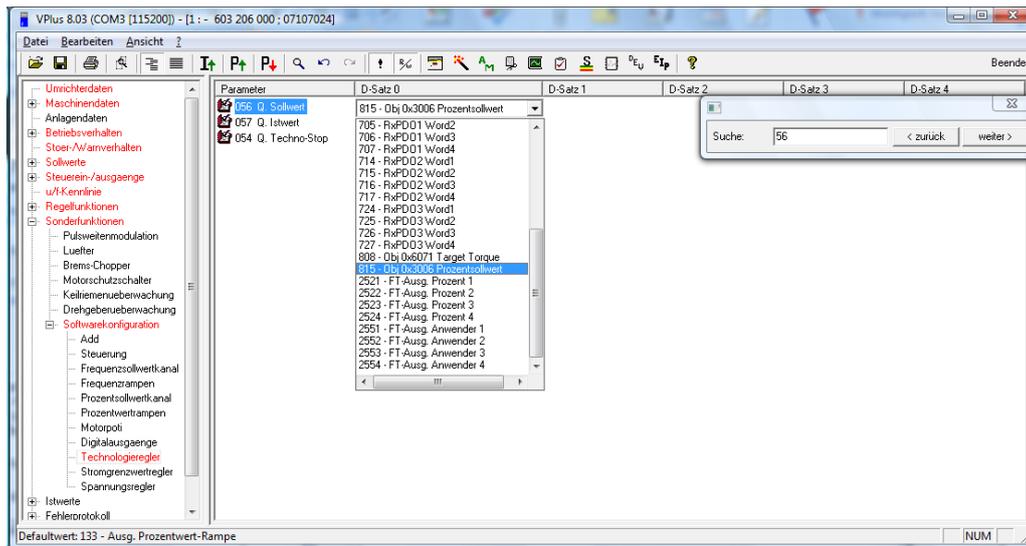
Via object 0x3006, you can write a percentage source – e.g. parameter *S. reference* **056**.

The value of object 0x3006 is available as a source and can be selected as 815 – Obj 0x3006 Reference Percentage via a choice list for parameters.

The values of object 0x3006 range from -30000 to 30000 (equivalent to percentage range from -300.00 % to 300.00 %)

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

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



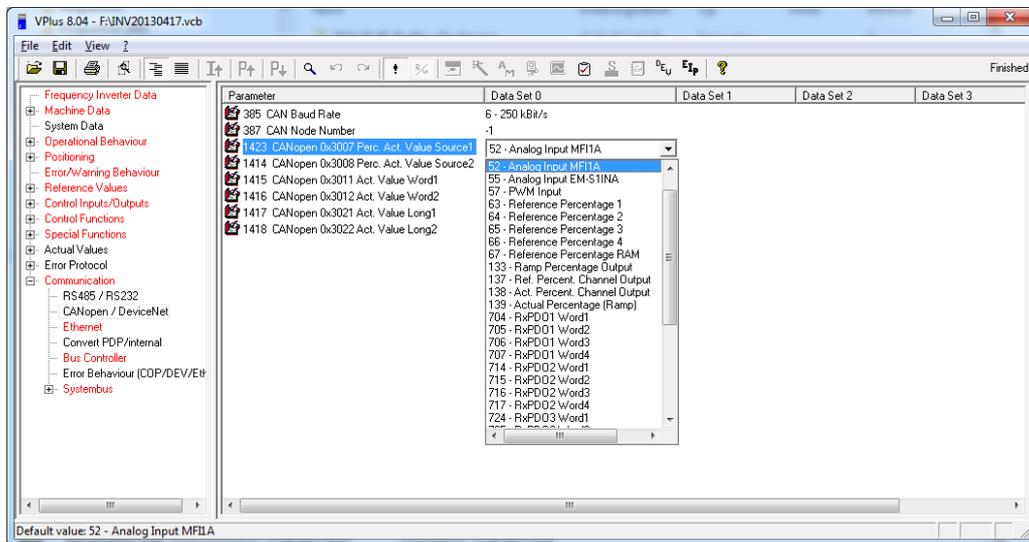
The percentage value is scaled as $\text{percent} * 100$ (e.g. 5678 represents 56.78%).

10.3.4.7 0x3007/0 Percentage Actual Value Source 1

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3007	0	Percentage Actual Value Source 1	Unsigned16	ro	Tx	

Object 0x3007 shows the actual value of the percentage source which can be selected via parameter *CANopen Percentage Actual Value Source 1* **1423**.

Default setting: 52 – Analog input MF1IA.



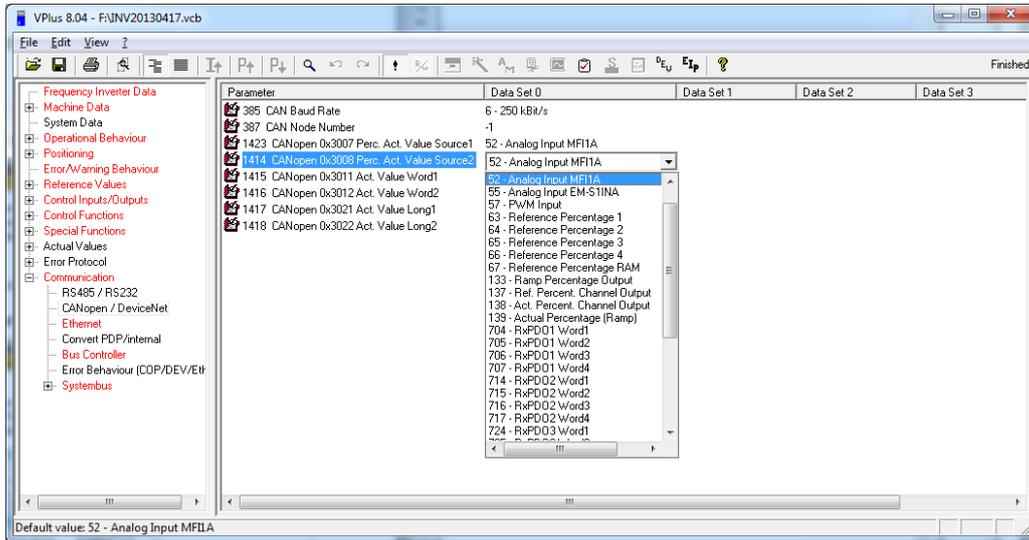
The percentage value is scaled as $\text{percent} * 100$ (e.g. 5678 represents 56.78%).

10.3.4.8 0x3008/0 Percentage Actual Value Source 2

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x3008	0	Percentage Actual Value Source 2	Unsigned16	ro	Tx	

Object 0x3008 shows the actual value of the percentage source which can be selected via parameter *CANopen Percentage Actual Value Source 2* **1414**.

Default setting: 52 – Analog input MF1IA.



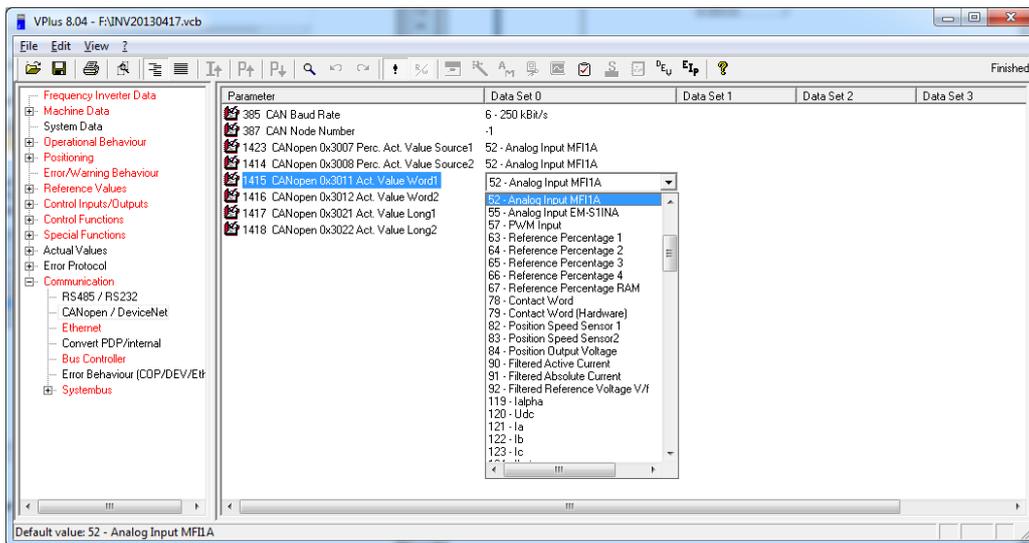
The percentage value is scaled as $\text{percent} * 100$ (e.g. 5678 represents 56.78%).

10.3.4.9 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 shows the actual value of the word source which can be selected via parameter *CANopen 0x3011 Act. Value Word 1* **1415**.

Default setting: 52 - Analog input MF1IA.

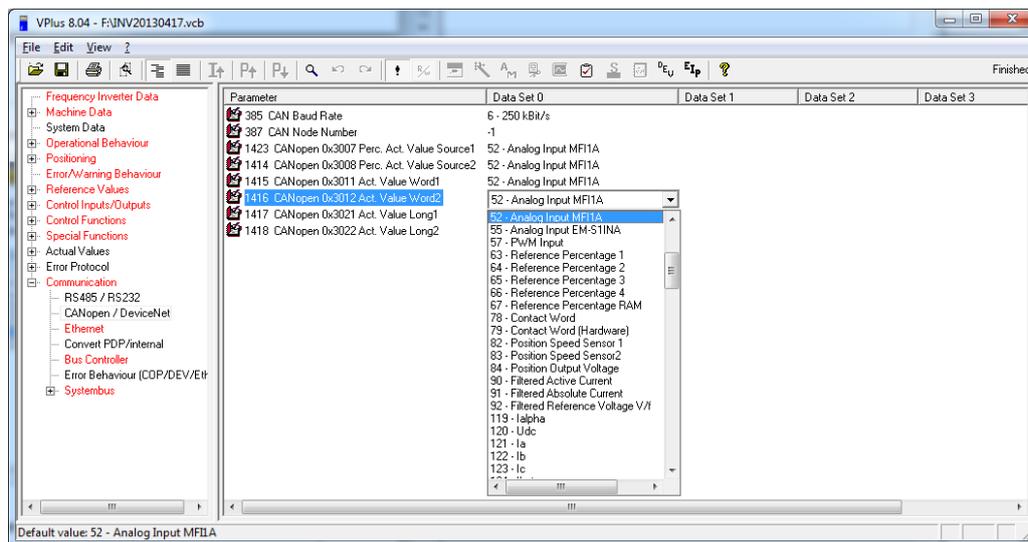


10.3.4.10 0x3012/0 Actual Value Word 2

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

Object 0x3012 shows the actual value of the word source which can be selected via parameter *CANopen 0x3012 Act. Value Word 2* **1416**.

Default setting: 52 - Analog input MF1IA.

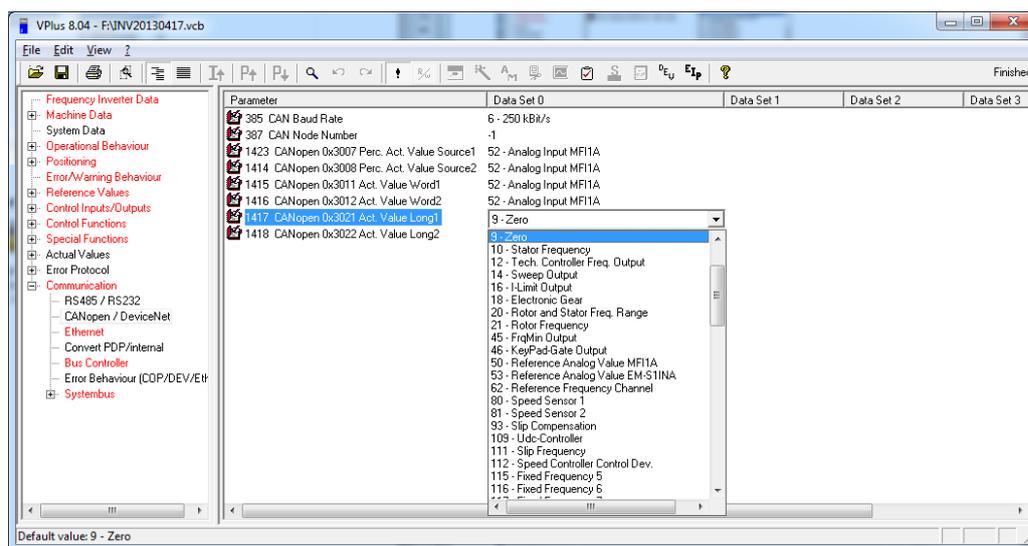


10.3.4.11 0x3021/0 Actual Value Long 1

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

Object 0x3021 shows the actual value of the long source which can be selected via parameter *CANopen 0x3021 Act. Value Long 1* **1417**.

Default setting: 9-zero.

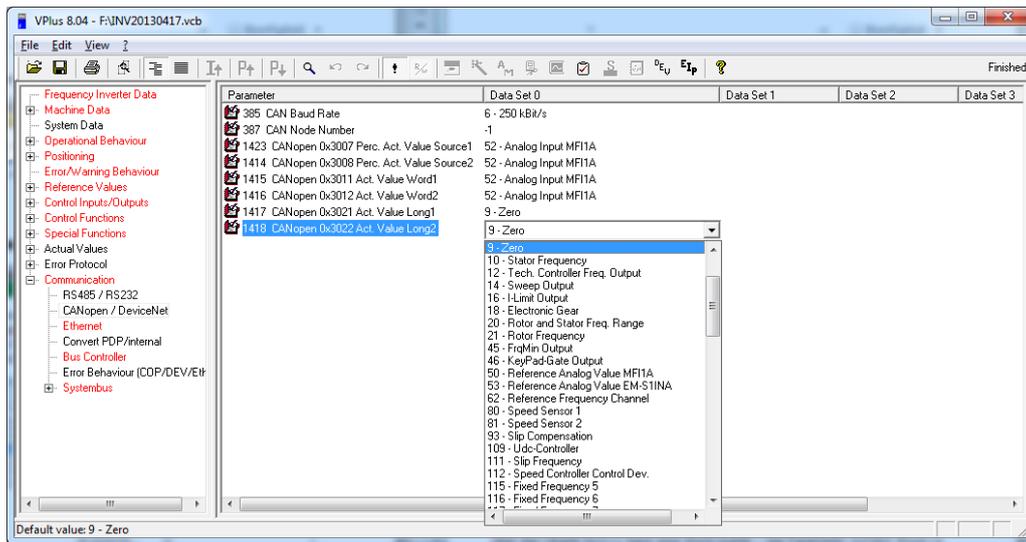


10.3.4.12 0x3022/0 Actual Value Long 2

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

Object 0x3022 shows the actual value of the long source which can be selected via parameter *CANopen 0x3022 Act. Value Long 2* **1418**.

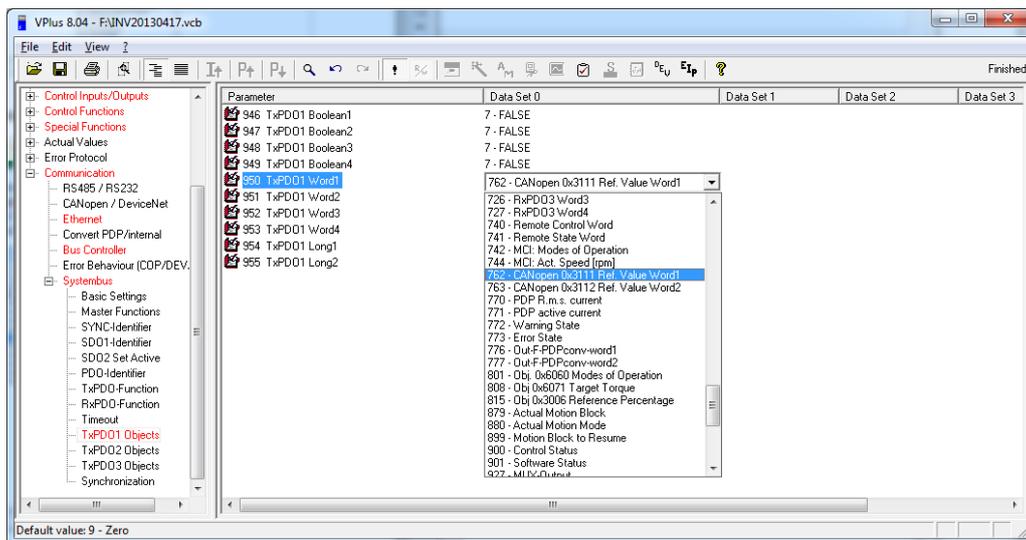
Default setting: 9-zero.



10.3.4.13 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

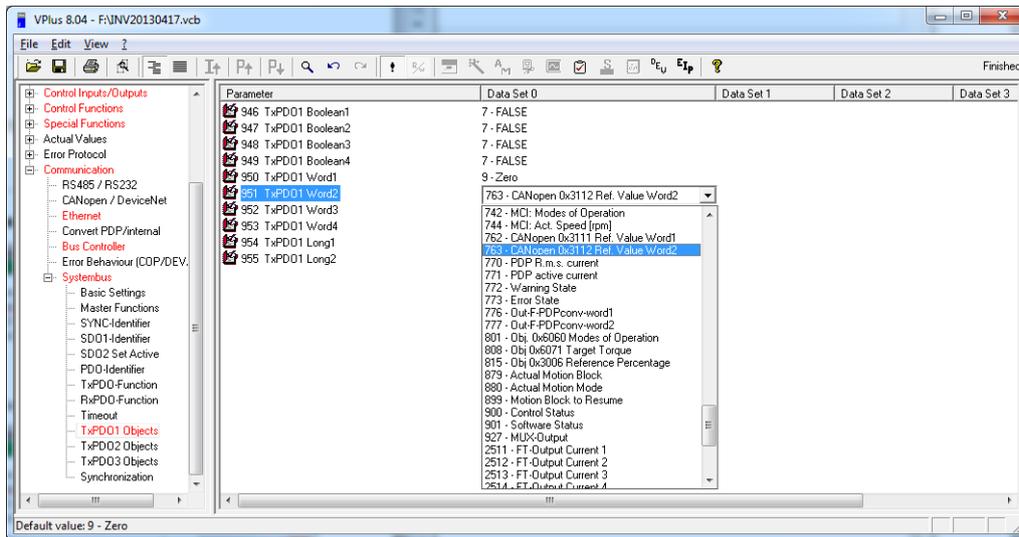
Via object 0x3111, you can write a word source – e.g. parameter *TxPDO1 Word 1* **950** of Systembus. The value of object 0x3111 is available as a source and can be selected as 762 – CANopen 0x3111 Ref. Value via a choice list for parameters.



10.3.4.14 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

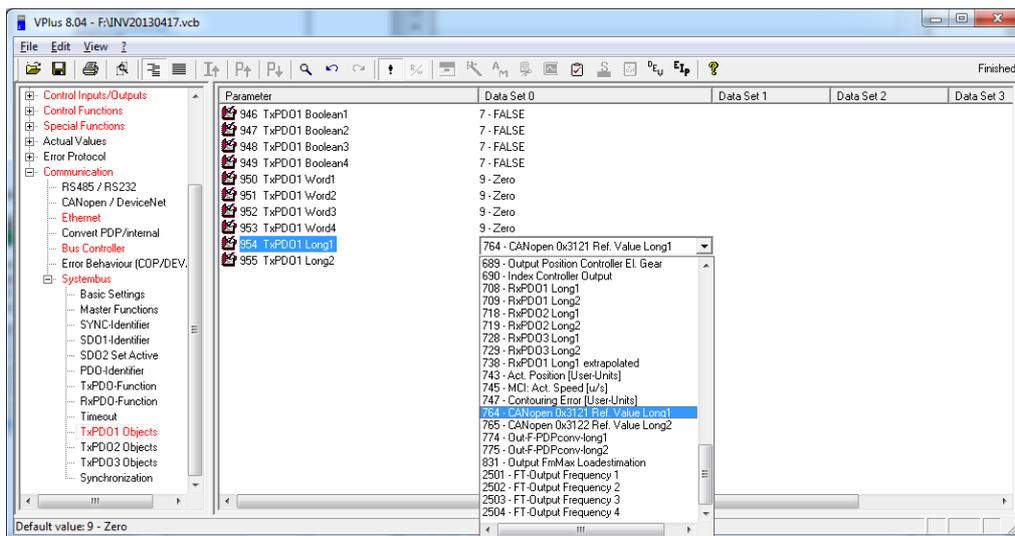
Via object 0x3112, you can write a word source – e.g. parameter *TxPDO1 Word 2* **951** of Systembus. The value of object 0x3112 is available as a source and can be selected as 763 – CANopen 0x3112 Ref. Value via a choice list for parameters.



10.3.4.15 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

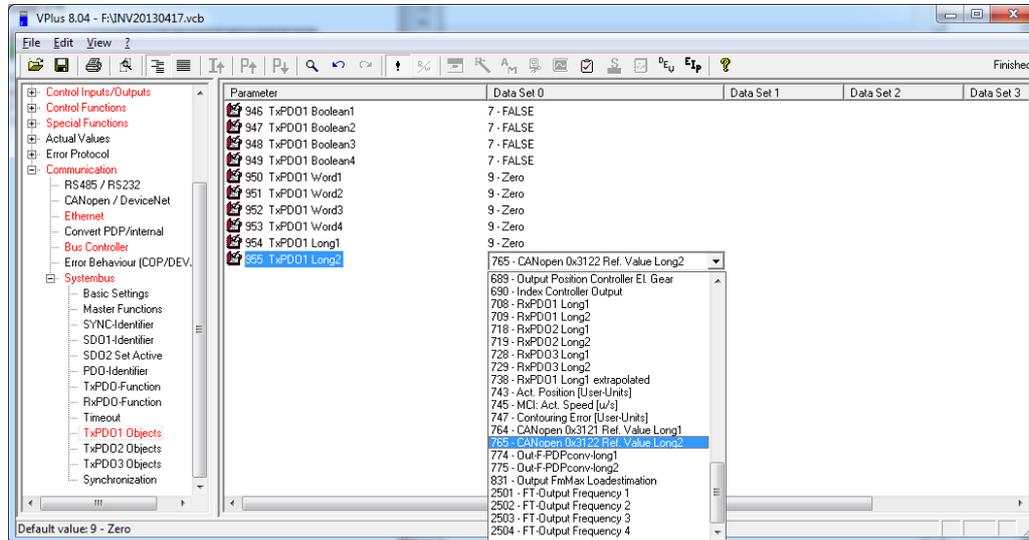
Via object 0x3121, you can write a long source – e.g. parameter *TxPDO1 Long 1* **954** of Systembus. The value of object 0x3121 is available as a source and can be selected as 764 – CANopen 0x3121 Ref. Value via a choice list for parameters.



10.3.4.16 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, you can write a long source – e.g. parameter *TxPDO1 Long 2* **955** of Systembus. The value of object 0x3122 is available as a source and can be selected as 765 – CANopen 0x3122 Ref. Value via a choice list for parameters.



10.3.4.17 0x5F10/n 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

<p>Object works in:</p> <ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ Electronic Gear: Slave ○ Table Travel Record mode (Electronic Gear operation) 	<p>Object does not work in:</p> <ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ Profile Positioning mode ○ Velocity mode ○ Profile Velocity mode ○ Homing mode ○ 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** = x40). 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.

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).

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.
--------	--

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

10.3.4.18 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Electronic Gear: Slave o Table Travel Record mode (Electronic Gear operation) 	<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing 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** = x40). 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 correspondingly.

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*.

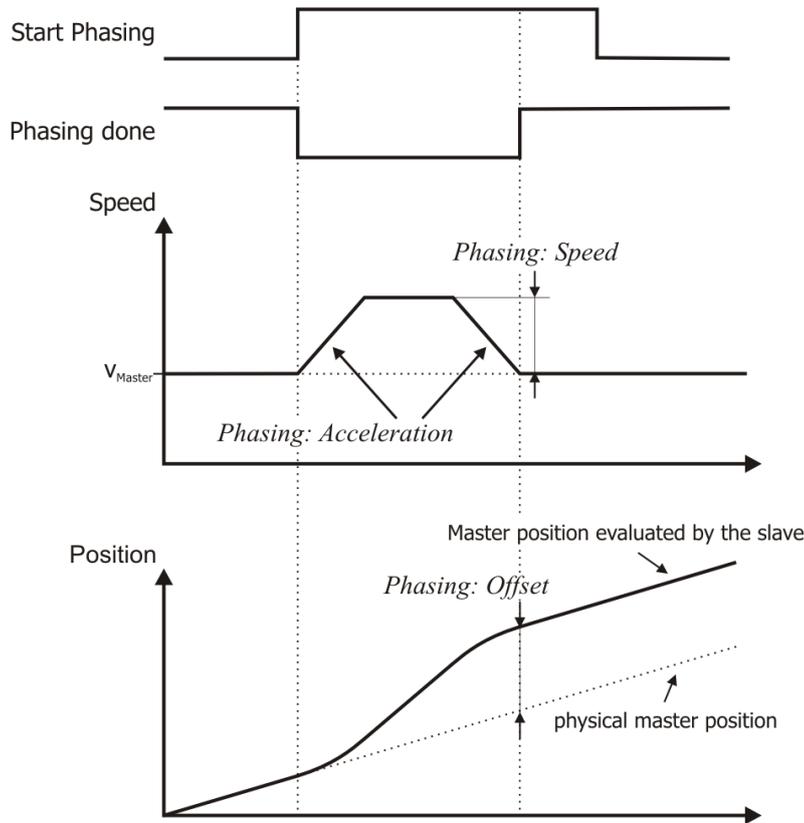
The function is started via Bit 9 of the Control Word. After start, 0x5F11/2 *Phasing 1: Speed* and 0x5F11/3 *Phasing 1: Acceleration* are used until the slave position is offset from the master position by *Phasing 1: Offset*.

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 profiles 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)
0	1	2 (0x5F12)
1	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 0x5F12/1 0x5F13/1 0x5F14/1	Phasing 1: Offset Phasing 2: Offset Phasing 3: Offset Phasing 4: Offset 1125.1 1125.2 1125.3 1125.4 <i>Phasing: Offset</i>
0x5F11/2 0x5F12/2 0x5F13/2 0x5F14/2	Phasing 1: Speed Phasing 2: Speed Phasing 3: Speed Phasing 4: Speed 1126.1 1126.2 1126.3 1126.4 <i>Phasing: Speed</i>
0x5F11/3 0x5F12/3 0x5F13/3 0x5F14/3	Phasing 1: Acceleration Phasing 2: Acceleration Phasing 3: Acceleration Phasing 4: Acceleration 1127.1 1127.2 1127.3 1127.4 <i>Phasing: Acceleration</i>

10.3.4.19 0x5F15/0 In Gear Threshold

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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Electronic Gear: Slave o Table Travel Record mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing 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](#) .



When 0x5F15/0 In Gear Threshold is set to zero, the signal "in gear" is set as soon as the drive reaches 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 v/velocity max amount](#) or *Maximum frequency 419*. It is set either via [0x6046/2 v/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	
No.	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	" <i>In-Gear</i> "-Threshold

10.3.4.20 0x5F16/0 In Gear Time

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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Electronic Gear: Slave o Table Travel Record mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing 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 *vl velocity max amount* or *Maximum frequency 419*. It is set either via 0x6046/2 *vl 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 does not affect the Signal "In Gear".

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

Object		Setting	
No.	Object	Min.	Max.
	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	<i>In Gear-Time</i>

10.3.4.21 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 does not work in:
<ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ All modes 	<ul style="list-style-type: none"> – 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.

NOTICE

Risk of component damage!

The Output of the Position Controller is not limited by [0x6046/2 vl 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.

- Observe all relevant settings carefully.

*Maximum speed** and *Limitation 1118* must be set for fitting values during the commissioning.

Chapter **18.4** 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](#) v/ *velocity max amount* or *Maximum frequency 419*. It is set either via [0x6046/2](#) v/ *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	
No.	Object	Min.	Max.
0x5F17/1	Position Controller: Time Constant	1,00 ms	300,00 ms
0x5F17/2	Position Controller: Limitation	0	2147483647 (= 0x7FFF FFFF)

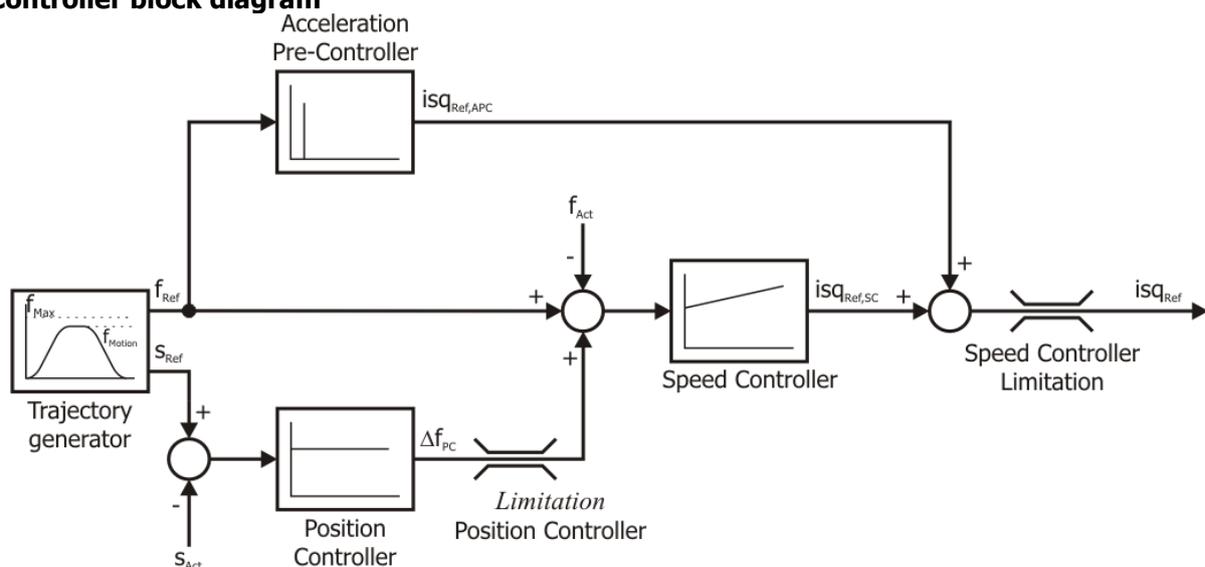
Alternatively the parameters **1104** and **1118** can be used instead of the Objects.

Object		Parameter	
0x5F17/1	Position Controller: Time Constant	1104	<i>Time Constant</i>
0x5F17/2	Position Controller: Limitation	1118	<i>Limitation</i>

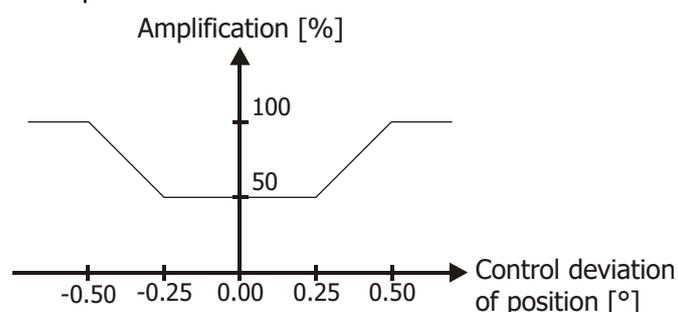
Example:

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 behavior 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 behavior in all situations.

10.3.4.22 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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch o 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 12.4.7.2 "Master/Slave Position Correction".



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	<i>M/S Synchronization Offset</i>

10.3.4.23 0x5FF0/0 Active motion block

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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing 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** = x40). 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.

10.3.4.24 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	

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Profile Velocity mode o Homing 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** = x40). 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.

10.3.5 Device Profile Objects (0x6nnn)

10.3.5.1 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* defines the operating characteristics of the frequency inverter in the case of an error in the bus connection due to BusOff, RxPDO length error or NMT state change (leaving of NMT state "Operational").

Depending on the setting of *Local/Remote* **412**, the response of the setting of object 0x6007 will change as shown in the following table.

Object 0x6007/0		
Operation mode	Function with "Control via Statemachine"	Function in "Other control"
0 - No reaction	Operating point is maintained.	Operating point is maintained.
1 - Error	"Fault" status will be activated immediately. Factory setting.	The controller (state machine) switches to "Fault" state immediately.
2 - Stop	Control command "Disable voltage" and switch to "switch on disabled" status.	
3 - Quick-Stop	Control command "Quick stop" and switch to "switch on disabled d" status.	
-1 - Ramp-Stop + (Minus 1) Error	Control command "Disable operation" and switch to "Error" status once the drive has been shut down.	
-2 - Quick-Stop + (Minus 2) Error	Control command "Quick stop" and switch to "Error" status once the drive has been shut down.	

NOTICE

Object *abort connection option code* corresponds to frequency inverter parameter *Bus Error Behavior* **388**.

The parameter settings *Bus Error Behavior* **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 behavior 388	0x6007
0	0
1	1
2	2
3	3
4	-1
5	-2



Writing of parameters *Bus Error Behavior* **388** and writing of object 0x6007 have the same effect.



When object 0x6007 was written and a parameter saving instruction (Object [0x1010](#)) was generated after that, the value of 0x6007 will be saved in the non-volatile memory. When the frequency inverter is switched on the next time, the value for 0x6007 set before will be activated again and overwrite the setting of Parameter *Bus Error Behavior* **388**.

For details about possible faults, refer to Chapter 18.4 "Error messages".

10.3.5.2 0x603F/0 Error code

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

In object *error code*, the last error that has occurred will be saved.

According to CANopen® DS402, a great number of possible error messages is specified. The following list shows the relation between the error code displayed by the frequency inverter on the control panel and the error saved in object *error code*.

Error reports			
Device Error	DS402 Error code	Meaning	
F00	xx 00 00	No error has occurred	
Overload			
F01	xx 23 10	Frequency inverter was overloaded	

Case				
F02	xx	42	10	Case temperature outside the temperature limits
Inside				
F03	xx	41	10	Inside temperature outside of 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 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 messages

If CANopen® DS402 *error code* 1000 = generic-error occurs, the error code can be read via parameter *Current error* **260** (unsigned16). Parameter *Current error* **260** contains the error code in internal product-specific format.

For the error code assignment table of the relevant messages, refer to 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.

10.3.5.3 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 *controlword* is relevant to the frequency inverter if parameter *Local/Remote* **412** is set to "1 - Control via statemachine". Object 0x6040/0 *controlword* is linked to the internal parameter *Control word* **410**. When using CANopen® use object 0x6040/0 *controlword* instead of parameter *Control word* **410**.

Control word (Control word)	
Bit	
0	Switch on
1	Enable voltage
2	Quick stop (Low active)
3	Enable operation
4	Operation mode specific
5	Operation mode specific
6	Operation mode specific
7	Fault reset
8	Halt
9	Operation mode specific
10	Reserved
11	Manufacturer specific
12	Manufacturer specific
13	Manufacturer specific
14	Manufacturer specific
15	Manufacturer specific



Bits 4, 5, 6 and 9 ... 15 are used in motion control configurations (**P.30** = $\times 40$) only. See chapter 12 "Control of frequency inverter" and 18.1 "Control Word/Status word Overview".

10.3.5.4 0x6041/0 Status word

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

Object 0x6041/0 *statusword* shows the current state of the frequency inverter.

Object 0x6041/0 *statusword* is linked to the internal parameter *Status word* **411**.

When using CANopen® use object 0x6041/0 *statusword* instead of parameter *Status word* **411**.

Object 0x6041/0 Status word

Bit	
0	Ready to switch on
1	Enable voltage
2	Switched on
3	Enable operation
4	Operation enabled
5	Operation mode specific
6	Fault
7	Fault reset
8	Voltage enabled
9	Operation mode specific
10	Quick stop (Low active)
11	Manufacturer specific
12	Switch on disabled
13	Manufacturer specific
14	Warning
15	Manufacturer specific Warning2



Bits 8, 12, 13 and 14 are used in motion control configurations (**P.30** = $\times 40$) only. See chapter 12 "Control of frequency inverter" and 18.1 "Control Word/Status word Overview".

10.3.5.5 0x6042/0 v/target velocity [rpm]

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

Object works in:	Object does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode - Non motion Control (conf. $\neq \times 40$) 	<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode o Profile Velocity mode o Profile Positioning mode o Homing mode o Move away from Limit Switch o Electronic Gear: Slave

The object *vl target velocity* is the reference speed for the frequency inverter. *vl target velocity* is interpreted as a rotational speed with unit min^{-1} . The internal reference frequency of the frequency inverter is calculated from the target velocity in min^{-1} considering parameter *No. of pole pairs* **373**.

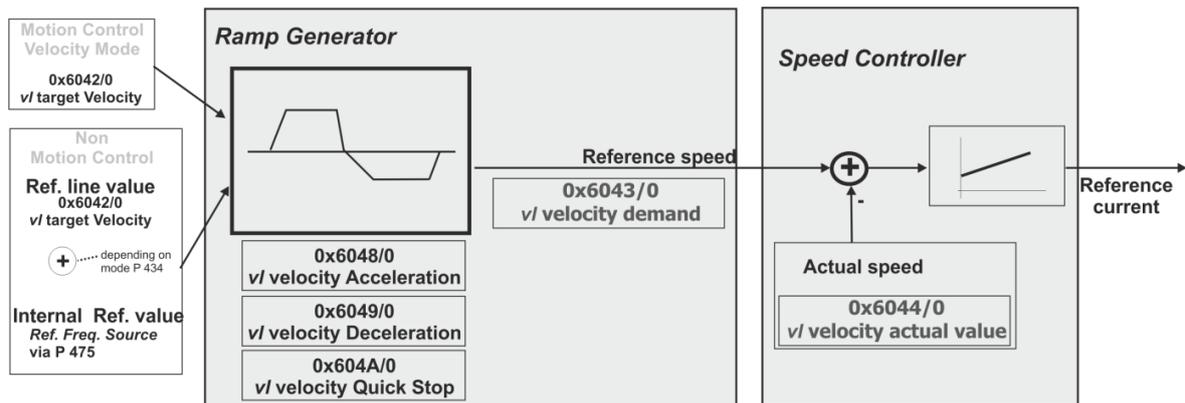


Parameter *No. of pole pairs* **373** has four different data sets. In applications with Motion control (configuration = $\times 40$), only data set 1 is used.

In applications without Motion Control (configuration $\neq \times 40$), more than one motor is connected to the frequency inverter in many cases (only one motor will be active at a time, switched via contactor). These motors can have different numbers of pole pairs. In this case, the entry in parameter *No. of pole pairs* **373** will be different in the four datasets. Upon switching to a motor, object *v/ target velocity* must be written at least once in order to enable calculation of the reference frequency of the frequency inverter with the right number of pole pairs.

Parameter		Setting	
No.	Object	Min.	Max.
0x6042	v/ target velocity	-32768	32767

In Non motion Control configurations (conf. $\neq \times 40$) the *v/ 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 12.3.3 "Reference value / actual value").



10.3.5.6 0x6043/0 v/velocity demand [rpm]

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

Object *v/ velocity demand* is the output quantity of the ramp function in unit min^{-1} . The object has the same notation as object *v/ target velocity* and can be read as an actual value. For calculation of *v/ velocity demand*, parameter *No. of pole pairs* **373** is considered (in the same way as described for object *v/ target velocity*).

10.3.5.7 0x6044/0 v/velocity actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6044	0	v/ velocity actual value	Integer16	ro	Tx	

Object *v/ velocity actual value* is the current speed of the drive in min^{-1} . The object has the same notation as object *v/ target velocity* and can be read as an actual value. For calculation of *v/ velocity actual value demand*, parameter *No. of pole pairs* **373** is considered (in the same way as described for object *v/ target velocity*).

10.3.5.8 0x6046/n v/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	v/velocity min amount (RPM)	Unsigned32	rw	No	See text
	2	v/velocity max amount (RPM)	Unsigned32	rw	No	See text

Object works in:	Object does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes - Non motion Control (conf. \neq x40) 	

Object vl velocity min max amount comprises Subindex 01 = vl velocity min amount and Subindex 02 = vl velocity max amount.

The unit of 0x6046/n vl velocity min max amount is min^{-1} (positive values only). The value of 0x6046/n vl velocity min max amount is converted to a frequency value internally, considering parameter *No. of pole pairs* **373** (in dataset 1).

Writing of object 0x6046/1 vl velocity min amount will automatically generate a write instruction for parameter *Minimum frequency* **418** to RAM (\rightarrow dataset 5, all datasets in RAM only).

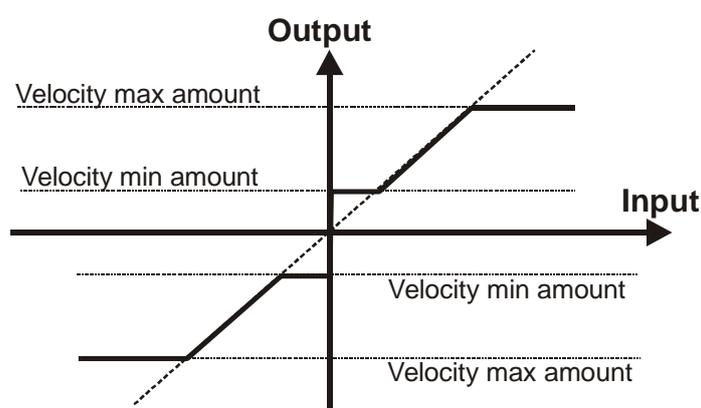
Writing of object 0x6046/2 vl velocity max amount will automatically generate a write instruction for parameter *Maximum frequency* **419** to RAM (\rightarrow dataset 5, all datasets in RAM only).

The default setting depends on the motor settings used.



When the reference value specified with object [0x6042 vl target velocity](#) is smaller than object value 0x6046/1 vl velocity min amount or greater than 0x6046/2 vl velocity max amount, [0x6042 vl target velocity](#) is limited to the relevant values.

No.	Object	Min.	Max.
0x6046/1	vl/velocity min amount (RPM)	1	32767 (= 0x7FFF)
0x6046/2	vl/velocity max amount (RPM)	1	32767 (= 0x7FFF)



When objects 0x6046/1 or 0x6046/2 are written and a save parameters instruction is generated after that (object [0x1010](#)), the object values in the non-volatile memory will be saved. When the frequency inverter is switched on the next time, the values set before will be activated again 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**.

10.3.5.9 0x6048/n vl/velocity acceleration

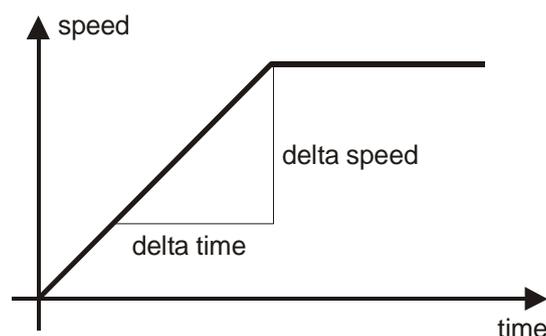
Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6048	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode - Non motion Control (conf. ≠ x40) 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode o Profile Velocity mode o Profile Positioning mode o Homing mode o Move away from Limit Switch o Electronic Gear: Slave
---	--

With object 0x6048 *v/ velocity acceleration* the change of speed and startup time is set in **velocity mode**. Object 0x6048 *v/ velocity acceleration* comprises *delta speed* in min^{-1} and *delta time* in seconds. The frequency gradient during startup is written to parameters *Acceleration clockwise* **420** and *Acceleration anticlockwise* **422** (dataset 5, all datasets only in RAM). Both parameters are set to the same value. The values of parameters *Acceleration clockwise* **420** and *Acceleration anticlockwise* **422** are converted to a value in unit frequency/second, considering parameter *No. of pole pairs* **373** (in dataset 1).

The gradient is changed internally by the change in 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)



When objects 0x6048/1 or 0x6048/2 are written and a save parameters instruction is generated after that (object [0x1010](#)), the object values in the non-volatile memory will be saved. When the frequency inverter is switched on the next time, the values set before will be activated again and overwrite the settings of parameters *Acceleration clockwise* **420** and *Acceleration anticlockwise* **422**.

10.3.5.10 0x6049/n *v/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 (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

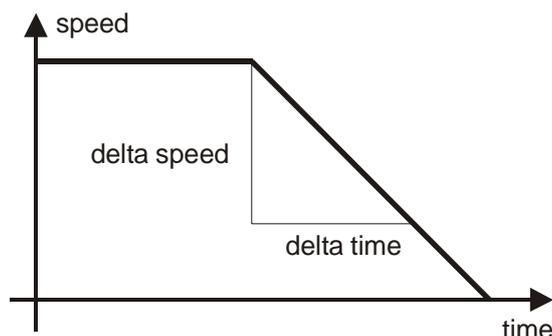
<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode - Non motion Control (conf. ≠ x40) 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode o Profile Velocity mode o Profile Positioning mode o Homing mode o Move away from Limit Switch o Electronic Gear: Slave
---	--

With object 0x6049 *v/ velocity deceleration* the change of speed and shutdown time is set. Object 0x6049 *v/ velocity deceleration* comprises *delta speed* in min^{-1} and *delta time* in seconds.

The frequency gradient during shutdown is written to parameters *Deceleration clockwise* **421** and *Deceleration anticlockwise* **423** (dataset 5, all datasets only in RAM). Both parameters are set to the same value. The values of parameters *Deceleration clockwise* **421** and *Deceleration anticlockwise* **423** are converted to a value in unit frequency/second, considering parameter *No. of pole pairs* **373** (in dataset 1).

The gradient is changed internally by the change in 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)



When objects 0x6049/1 or 0x6049/2 are written and a save parameters instruction is generated after that (object [0x1010](#)), the object values in the non-volatile memory will be saved. When the frequency inverter is switched on the next time, the values set before will be activated again and overwrite the settings of parameters *Deceleration clockwise* **421** and *Deceleration anticlockwise* **423**.

10.3.5.11 0x604A/n v/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 (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode - Non motion Control (conf. \neq x40) 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Table Travel record mode o Profile Velocity mode o Profile Positioning mode o Homing mode o Move away from Limit Switch o Electronic Gear: Slave
---	--

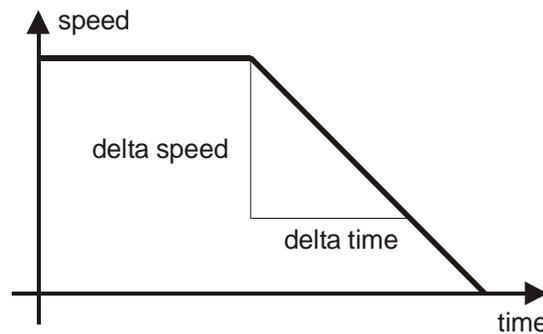
With object 0x604A *v/velocity quick stop*, you can set the quick stop deceleration. Object 0x604A *v/velocity quick stop* comprises speed change in min^{-1} and shutdown time in seconds.

The frequency gradient during shutdown is written to parameters *Emergency Stop Clockwise* **424** and *Emergency Stop Anticlockwise* **425** (dataset 5, all datasets only in RAM). Both parameters are set to the same value. The value of parameters *Emergency Stop Clockwise* **424** and *Emergency Stop Anticlockwise* **425** are converted internally to a value in unit frequency/second, considering parameter *No. of pole pairs* **373** (in dataset 1).

The gradient is changed internally by the change in objects *delta-time* or *delta-speed*.

Parameter		Setting	
No.	Object	Min.	Max.
0x604A/1	Delta speed (rpm)	1	32767 (= 0x7FFF)

0x604A/2	Delta time (sec)	1	65535 (= 0xFFFF)
----------	------------------	---	---------------------



When objects 0x604A/1 or 0x604A/2 are written and a save parameters instruction is generated after that (object [0x1010](#)), the object values in the non-volatile memory will be saved. When the frequency inverter is switched on the next time, the values set before will be activated again and overwrite the settings of parameters *Emergency Stop Clockwise* **424** and *Emergency Stop Anticlockwise* **425**.

10.3.5.12 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<ul style="list-style-type: none"> - Non motion Control (conf. ≠ x40)

With object **0x6060** *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 (**P.30** = x40 and **412** *Local/Remote* = "1 – Control via Statemachine"):

<i>Modes of operation</i>		
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 - not supported by EtherNet/IP™	
8 0x08	- Cyclic sync position mode - not supported by EtherNet/IP™	
9 0x09	- Cyclic sync velocity mode - not supported by EtherNet/IP™	
-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 as 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 (**P.30** = $\times 40$ and **412** *Local/Remote* = "1 – Control via Statemachine")

<i>Modes of operation</i>	
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 12 "Control of frequency inverter".

10.3.5.13 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 does not work in:
<ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ All modes 	<ul style="list-style-type: none"> – Non motion Control (conf. $\neq \times 40$) <ul style="list-style-type: none"> ○ 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 12 "Control of frequency inverter".

10.3.5.14 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 does not work in:
<ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ All modes 	<ul style="list-style-type: none"> – Non motion Control (conf. $\neq \times 40$)

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**.

10.3.5.15 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 does not work in:
<ul style="list-style-type: none"> – Motion Control: <ul style="list-style-type: none"> ○ All modes 	<ul style="list-style-type: none"> – Non motion Control (conf. $\neq \times 40$)

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](#).

10.3.5.16 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<ul style="list-style-type: none"> - 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*.

10.3.5.17 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<ul style="list-style-type: none"> - Non motion Control (conf. ≠ x40)

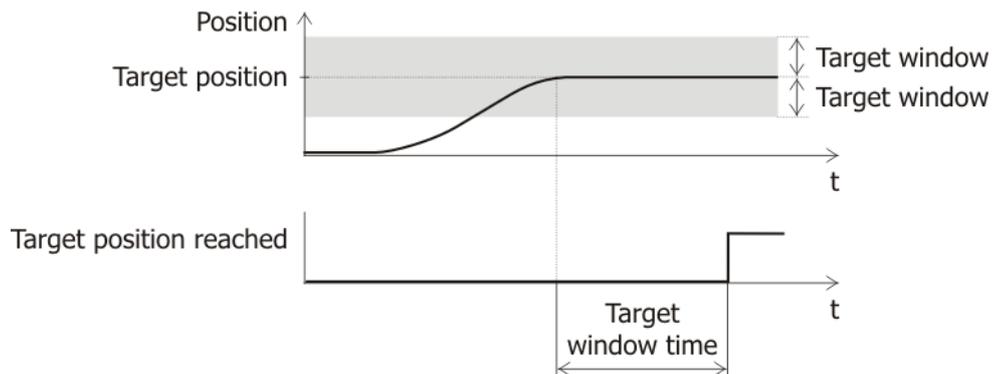
The signal "target position reached" can be changed in accuracy with Object 0x6067 *Position window* for the modes which use Status Word Bit 10 "Target reached" as "Target Position reached" like "Profile Positioning Mode" and "Table Travel Record Mode".

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*.

10.3.5.18 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<ul style="list-style-type: none"> - 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**.

10.3.5.19 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<ul style="list-style-type: none"> - Non motion Control (conf. ≠ x40)

The actual velocity value in [u/s] is displayed.

10.3.5.20 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 does not work in:
<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode 	<ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Homing 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. It is recommended to set 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](#).

10.3.5.21 0x606E/0 Velocity Window Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606E	0	Velocity Window time	Unsigned16	rw	No	0

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Homing 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](#).

10.3.5.22 0x606F/0 Velocity Threshold

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x606F	0	Velocity Threshold	Unsigned16	rw	No	100

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Homing 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 μ 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](#).

10.3.5.23 0x6070/0 Velocity Threshold Time

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6070	0	Velocity Threshold Time	Unsigned16	rw	No	0

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch o Electronic Gear: Slave - Non motion Control (conf. \neq 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](#).

10.3.5.24 0x6071/0 Target Torque

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6071	0	Target Torque	Integer16	rw	Rx	0

The value that can be transmitted via object 0x6071 is selectable as source 808 for various parameters (e.g. *FT input buffer percentage* **1381**).

It is also available as operation mode 95 or inverted operation mode 195 (e.g. for parameter *Reference percentage source* **476**) in configurations with torque control (*Configuration* **30** = x30).

The value 0x3E8 (=1000) is the rated motor torque (100.0 %).



By default, object 0x6071 is not linked to a device function. In order to use object 0x6071, at least one device function must be linked to the object by setting the relevant parameters.

The values of object 0x6071 range from -3000 to 3000 (= -300.0...300.0 %).

Parameter		Setting	
No.	Object	Min.	Max.
0x6071/0	Target Torque	-3000 (= 0xF448)	3000 (= 0x0BB8)

Hexadecimal value 0x6071	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

10.3.5.25 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* shows the actual torque (see parameter *Torque* **224**).

The value 0x3E8 (=1000) is the rated motor torque (100.0 %). Please also note object [0x6071](#).

10.3.5.26 0x6078/0 Current actual value

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6078	0	Current actual value	Integer16	ro	Tx	

Object 0x6078 *Current actual value* shows the actual current value (see parameter *R.m.s current* **211**).

The value 0x3E8 (=1000) is the rated motor current (100.0 %). The rated motor current is set during commissioning in parameter *Rated current* **371**.

10.3.5.27 0x6079/0 DC link 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* shows the actual value of the DC link voltage in mV (see parameter *DC link voltage* **222**).

Value 0x0001 86A0 (=100 000) is equivalent to 100.000 V (three decimal places).

10.3.5.28 0x607A/0 Target position

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x607A	0	Target position	Integer32	rw	Rx	0

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Profile Velocity mode o Homing 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](#).

10.3.5.29 0x607C/0 Home offset

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x607C	0	Target position	Integer32	rw	No	0

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Homing mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Profile Velocity mode o Profile Positioning 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 Offset 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](#).

10.3.5.30 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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch o Electronic Gear: Slave - Non motion Control (conf. \neq 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)

10.3.5.31 0x6083/0 Profile acceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6083	0	Profile acceleration	Unsigned32	rw	Rx	0x5 0000

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode o Profile Positioning mode o Electronic Gear: Slave 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch - Non motion Control (conf. \neq x40)
---	---

Object 0x6083 *profile acceleration* is the acceleration in user units per second² [u/s²] 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 ($2^{31}-1$).

Parameter		Setting	
No.	Object	Min.	Max.
0x6083/0	Profile acceleration (u/s ²)	1	2147483647 (= 0x7FFF FFFF)

10.3.5.32 0x6084/0 Profile deceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6084	0	Profile deceleration	Unsigned32	rw	Rx	0x5 0000

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Velocity mode o Profile Positioning mode o Electronic Gear: Slave 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch - Non motion Control (conf. \neq x40)
---	---

Object 0x6084 *profile deceleration* is the deceleration in u/s^2 .



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 ($2^{31}-1$).

Parameter		Setting	
No.	Object	Min.	Max.
0x6084/0	Profile deceleration (u/s^2)	1	2147483647 (= 0x7FFF FFFF)

10.3.5.33 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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Profile Velocity mode o Homing mode o Table Travel record mode o Move away from Limit Switch <ul style="list-style-type: none"> o Electronic Gear: Slave 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Non motion Control (conf. \neq x40)
--	--

Object 0x6085 *quick stop deceleration* is the deceleration (in user units per second²) 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 ($2^{31}-1$).

Parameter		Setting	
No.	Object	Min.	Max.
0x6085/0	Quick stop deceleration (u/s^2)	1	2147483647 (= 0x7FFF FFFF)

10.3.5.34 0x6086/0 Motion profile type

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6086	0	Motion profile type	Integer16	rw	No	3

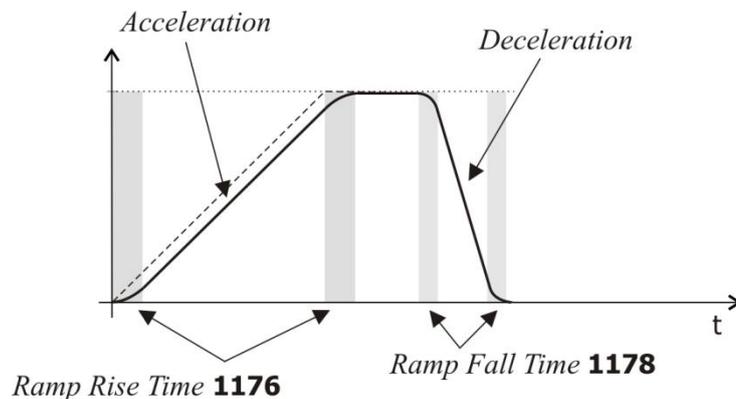
<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Profile Positioning mode o Profile Velocity mode o Move away from Limit Switch o Electronic Gear: Slave 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o Velocity mode o Homing mode o Table Travel record mode o Non motion Control (conf. ≠ x40)
---	---

Object 0x6086 *motion profile type* defines the ramp behavior 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.

10.3.5.35 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

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control: <ul style="list-style-type: none"> o All modes 	<p>Object does not work in:</p> <ul style="list-style-type: none"> o Non motion Control (conf. ≠ x40)
---	---

Object 0x6091 *gear ratio* defines the ratio of motor shaft revolutions to driving shaft revolutions.

$$\text{Gear ratio} = \frac{0x6091/1 \text{ motor shaft revolutions}}{0x6091/2 \text{ driving shaft revolutions}} \hat{=} \frac{\text{Parameter Gear Box : Motor Shaft Revolutions } \mathbf{1117}}{\text{Parameter Gear Box : Driving Shaft Revolutions } \mathbf{1116}}$$



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 <i>Gear Box: Motor Shaft Revolutions</i>
0x6091/2 Driving Shaft revolutions	1116 <i>Gear Box: Driving Shaft Revolutions</i>

The values of Objects 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)

10.3.5.36 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: – Motion Control : ○ All modes	Object does not work in: ○ Non motion Control (conf. ≠ x40)
--	---

Object 0x6092 *feed constant* defines the feed (in user units) per driving shaft revolutions.

$$\text{Feed constant} = \frac{0x6092/1 \text{ feed}}{0x6092/2 \text{ driving shaft revolu.}} \hat{=} \frac{\text{Parameter Feed Constant } \mathbf{1115}}{1}$$



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

10.3.5.37 0x6098/0 Homing method

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x6098	0	Homing method	Integer8	rw	No	0

<p>Object works in:</p> <ul style="list-style-type: none"> - Motion Control x40: <ul style="list-style-type: none"> o Homing mode 	<p>Object does not work in:</p> <ul style="list-style-type: none"> - Motion Control x40: <ul style="list-style-type: none"> o Profile Positioning mode o Profile Velocity mode o Velocity mode o Table Travel record mode o Move away from Limit Switch o Electronic Gear: Slave o 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.

Homing Method 0x6098/0		Function
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). 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.
8 -	Pos. Lim.-Sw., Ref.-Sig. right of left Edge of Home-Sw.	
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 <i>Home-Offset</i> 1131) is taken over as actual position value.

10.3.5.38 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
---	------------------------------	------------	----	----	----------

<p>Object works in:</p> <p>Motion Control:</p> <ul style="list-style-type: none"> ○ Homing mode ○ Move away from Limit Switch ○ Electronic Gear: Slave ¹⁾ 	<p>Object does not work in:</p> <p>Motion Control x40:</p> <ul style="list-style-type: none"> ○ Profile Positioning mode ○ Profile Velocity mode ○ Velocity mode ○ 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 12.4.7.2 "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*.

10.3.5.39 0x609A/0 Homing acceleration

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x609A	0	Homing acceleration	Unsigned32	rw	No	0x5 0000

<p>Object works in:</p> <p>Motion Control:</p> <ul style="list-style-type: none"> ○ Homing mode ○ Move away from Limit Switch ○ Electronic Gear: Slave¹⁾ 	<p>Object does not work in:</p> <p>Motion Control:</p> <ul style="list-style-type: none"> ○ Profile Positioning mode ○ Profile Velocity mode ○ Velocity mode ○ 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 12.4.7.2 "Master/Slave Position Correction".

Object 0x609A/0 *homing acceleration* defines acceleration and deceleration (in user units per second²) 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)

10.3.5.40 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	

<p>Object works in:</p> <p>Motion Control:</p> <ul style="list-style-type: none"> ○ All modes 	<p>Object does not work in:</p> <ul style="list-style-type: none"> ○ 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 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**.

10.3.5.41 0x60F8/0 Max Slippage [u/s]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60F8	0	Max Slippage	Integer32	rw	No	0

<p>Object works in: Motion Control:</p> <ul style="list-style-type: none"> ○ Profile Velocity mode 	<p>Object does not work in: Motion Control:</p> <ul style="list-style-type: none"> ○ Profile Positioning mode ○ Velocity mode ○ Homing mode ○ Table Travel record mode ○ Move away from Limit Switch ○ 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*.

10.3.5.42 0x60FF/0 Target Velocity [u/s]

Index	Sub-index	Meaning	Data type	Access	Map	Def.-Val
0x60F8	0	Target Velocity	Integer32	rw	Rx	0

<p>Object works in: Motion Control:</p> <ul style="list-style-type: none"> ○ Profile Velocity mode 	<p>Object does not work in: Motion Control:</p> <ul style="list-style-type: none"> ○ Profile Positioning mode ○ Velocity mode ○ Homing mode ○ Table Travel record mode ○ Move away from Limit Switch ○ Electronic Gear: Slave ○ Non motion Control (conf. ≠ x40)
---	---

Object 0x60FF *Target Velocity* defines the reference velocity in Profile velocity mode.

10.3.5.43 0x6502/0 Supported drive modes

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6502	0	Supported drive modes	Unsigned32	ro	No	

Object 0x6502 *Supported drive modes* displays the supported [0x6060](#) *Modes of Operation*.

The value 0x0000 0202 means:

- Bit 0: "Profile Positioning mode" is supported.
- Bit 1: "Velocity mode" is supported.
- Bit 2: "Profile Velocity mode" is supported.

- Bit 5: "Homing mode" is supported.
- Bit 6: "Interpolated Position mode" is **not supported**.
- Bit 7: "Cyclic Synchronous Positioning mode" is **not supported**.
- Bit 8: "Cyclic Synchronous Velocity mode" is **not supported**.

- Bit 16: Manufacturer mode -1 "Table Travel record mode" is supported.
- Bit 17: Manufacturer mode -2 "Move away from limit switch" is supported.
- Bit 18: Manufacturer mode -3 "Electronic Gear: Slave" is supported.

Index

A		
Acceleration		
Phasing	99	
Application objects	61	
B		
Boot-Up CANopen	67	
C		
Commissioning	22	
Communication objects.....	61, 77	
Communication Objects	84	
Configuration(s)	29	
Control word	107	
Copyright.....	8	
D		
Decommissioning	15	
Designated use	9	
Device profile objects	80, 105	
Disassembly		
Communication module	20	
E		
Electrical connections	13	
Emergency CANopen	65	
Error Codes CANopen	62	
EtherNet/IP™ overview.....	27	
F		
Font style.....	12	
G		
Gear ratio	124	
General Information about the Documentation	6	
Guarding CANopen	68	
H		
Heartbeat CANopen.....	69	
Homing		
Method.....	126	
Offset.....	121	
Speeds	127	
I		
Index Parameter	89	
Reading.....	90	
Writing	90	
Installation.....	13	
Communication module	19	
Instruction manuals	6	
IP-Settings	23	
M		
Manufacturer objects	78, 88	
Mapping		
RxPDO.....	86	
TxPDO	87	
Mapping CANopen		
RxPDO Communication	73	
TxPDO Communication	74	
Modes of operation	113	
N		
NMT CANopen	68	
NMT CANopen	66	
O		
Offset		
Phasing.....	99	
Operating behavior		
Bus connection failure.....	43	
OS Synchronization	25	
P		
Parameter access		
Reading Index Parameter.....	90	
Writing Index Parameter	90	
PDO CANopen	65	
Position deviation.....	103	
R		
Ramp Rise time	124	
S		
Safety		
General.....	9	
Safety function	15	
SDO CANopen	61	
Segmented Transfer CANopen	62	
Service.....	14	
Special CANopen Objects	69	
Status word	108	
Supported Classes/Instances/Attributes.....	30	
Sync CANopen	66	
V		
vl target velocity [rpm].....	108	
W		
Warranty and liability	7	

Bonfiglioli Worldwide Locations

Australia

Bonfiglioli Transmission (Aust.) Pty Ltd
2, Cox Place Glendenning NSW 2761
Locked Bag 1000 Plumpton NSW 2761
Tel. +61 2 8811 8000



Brazil

Bonfiglioli Redutores do Brasil Ltda
Travessa Cláudio Armando 171 - Bloco 3
CEP 09861-730 - Bairro Assunção
São Bernardo do Campo - São Paulo
Tel. +55 11 4344 2322



China

Bonfiglioli Drives (Shanghai) Co. Ltd.
#68, Hui-Lian Road, QingPu District,
201707 Shanghai
Tel. +86 21 6700 2000



France

Bonfiglioli Transmission s.a.
14 Rue Eugène Pottier
Zone Industrielle de Moimont II
95670 Marly la Ville
Tel. +33 1 34474510



Germany

Bonfiglioli Deutschland GmbH
Sperberweg 12 - 41468 Neuss
Tel. +49 0 2131 2988 0



Bonfiglioli Vectron GmbH

Europark Fichtenhain B6 - 47807 Krefeld
Tel. +49 0 2151 8396 0



O&K Antriebstechnik GmbH

Ruhrallee 8-12 - 45525 Hattingen
Tel. +49 0 2324 2050 1



India

Bonfiglioli Transmission Pvt. Ltd.
Mobility & Wind Industries
AC 7 - AC 11 Sidco Industrial Estate
Thirumudivakkam Chennai - 600 044
Tel. +91 844 844 8649



Discrete Manufacturing & Process Industries - Mechatronic & Motion

Survey No. 528/1
Perambakkam High Road Mannur Village,
Sriperumbudur Taluk Chennai - 602 105
Tel. +91 844 844 8649



Discrete Manufacturing & Process Industries

Plot No.A-9/5, Phase IV MIDC Chakan,
Village Nighoje Pune - 410 501
Tel. +91 844 844 8649



Italy

Bonfiglioli Riduttori S.p.A.
Discrete Manufacturing & Process Industries
Via Bazzane, 33/A
40012 Calderara di Reno
Tel. +39 051 6473111



Mobility & Wind Industries

Via Enrico Mattei, 12 Z.I. Villa Selva
47100 Forlì
Tel. +39 0543 789111



Discrete Manufacturing & Process Industries

Via Sandro Pertini lotto 7b
20080 Carpiano
Tel. +39 02985081



Bonfiglioli Mechatronic Research S.p.A

Via Unione 49 - 38068 Rovereto
Tel. +39 0464 443435/36



New Zealand

Bonfiglioli Transmission (Aust.) Pty Ltd
88 Hastie Avenue, Mangere Bridge,
2022 Auckland
PO Box 11795, Ellerslie
Tel. +64 09 634 6441



Singapore

Bonfiglioli South East Asia Pte Ltd
8 Boon Lay Way, #04-09,
8@ Tadehub 21, Singapore 609964
Tel. +65 6268 9869



Slovakia

Bonfiglioli Slovakia s.r.o.
Robotnícka 2129
Považská Bystrica, 01701 Slovakia
Tel. +421 42 430 75 64



South Africa

Bonfiglioli South Africa Pty Ltd.
55 Galaxy Avenue, Linbro Business Park,
Sandton, Johannesburg
2090 South Africa
Tel. +27 11 608 2030



Spain

Tecnotrans Bonfiglioli S.A
Pol. Ind. Zona Franca, Sector C,
Calle F, nº 6 - 08040 Barcelona
Tel. +34 93 447 84 00



Turkey

Bonfiglioli Turkey Jsc
Atatürk Organize Sanayi Bölgesi,
10007 Sk. No. 30
Atatürk Organize Sanayi Bölgesi,
35620 Çiğli - İzmir
Tel. +90 0 232 328 22 77



United Kingdom

Bonfiglioli UK Ltd.
Unit 1 Calver Quay, Calver Road, Winwick
Warrington, Cheshire - WA2 8UD
Tel. +44 1925 852667



USA

Bonfiglioli USA Inc.
3541 Hargrave Drive
Hebron, Kentucky 41048
Tel. +1 859 334 3333



Vietnam

Bonfiglioli Vietnam Ltd.
Lot C-9D-CN My Phuoc Industrial Park 3
Ben Cat - Binh Duong Province
Tel. +84 650 3577411



 PRODUCTION

 ASSEMBLY

 SALES

 SERVICE



Abbiamo un'inflessibile dedizione per l'eccellenza, l'innovazione e la sostenibilità. Il nostro Team crea, distribuisce e supporta soluzioni di Trasmissioni e Controllo di Potenza per mantenere il mondo in movimento

We have a relentless commitment to excellence, innovation & sustainability. Our team creates, distributes and services world-class power transmission & drive solutions to keep the world in motion.

Wir verpflichten uns kompromisslos zu Qualität, Innovation und Nachhaltigkeit. Unser Team entwickelt, vertreibt und wartet erstklassige Energieübertragungs- und Antriebslösungen, um die Welt in Bewegung zu halten

Notre engagement envers l'excellence, l'innovation et le développement durable guide notre quotidien. Notre Équipe crée, distribue et entretient des solutions de transmission de puissance et de contrôle du mouvement contribuant ainsi à maintenir le monde en mouvement.

Tenemos un firme compromiso con la excelencia, la innovación y la sostenibilidad. Nuestro equipo crea, distribuye y da soporte en soluciones de transmisión y control de potencia para que el mundo siga en movimiento.

COD. VEC1037R0

HEADQUARTERS
Bonfiglioli S.p.A
Registered Office: Via Bazzane, 33
40012 Calderara di Reno BO
Head office: Via Isonzo, 65/67/69
40033 Casalecchio di Reno BO
ITALY

tel: +39 051647 3111
www.bonfiglioli.com