Bonfiglioli
Vectron

Agile

CANopen Communication manual
Frequency inverter 230V / 400V

Bonfiglioli
power, control and green solutions

CONTENTS

1 General Information on the Documentation

This documentation describes the communication with *Agile* device series frequency inverters using the CANopen protocol. The modular hardware and software structure allows the user-friendly customization of the frequency inverters. Applications, which demand high functionality and dynamics can be comfortably implemented.

1.1 Instruction Manuals

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

Quick Start Guide
The "Quick Start Guide" brief instructions manual describes the basic steps for the mechanical and electrical installation of the frequency inverter. The guided commissioning supports you with the selection of the necessary parameters and the software configuration.

Operating Instructions
The Operating Instructions documents the complete functionality of the frequency inverter. The parameters necessary for specific applications for adaptation to the application and the extensive 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 subjects connected with the use of the frequency inverter is described specific to the application.

The documentation and further information can be requested from the local BONFIGLIOLI representative.

The following instruction manuals are available for the *Agile* device series:

Agile Operating Instructions	Frequency inverter functionality.
Agile Quick Start Guide	Installation und commissioning. Supplied with the device.
Communication Application Manuals	Communication via the RS485 Interface on the X21-Connection (RJ45): Instructions for Modbus and VABus. Communication via the X12.5 and X12.6 Control Terminals: Instructions for Systembus and CANopen®[1]. Communication via the Communication Modules: CM-232/CM-485: Instructions for Modbus and VABus. CM-CAN: Instructions for Systembus und CANopen®. CM-PDPV1: Instructions for Profibus-DP-V1
PLC Application Manual	Logical interconnections of digital signals. Functions for analog signals such as comparisons and mathematical functions. Graphical support for the programming of functional components.
Service Instructions	For service personnel. Service work, monitoring of service intervals and replacement of ventilators.

This documentation has been produced with the greatest of care and extensively and repeatedly checked. For reasons of clarity, not all the detailed information on all types of the product and also not every imaginable case of installation, operation or maintenance has been taken into account. If you require further information or if specific problems which are not dealt with extensively enough in the documentation exist, you can request the necessary information from the local BONFIGLIOLI representative.

[1] The CANopen®-Communication products fulfill the specifications of the CiA® (CAN in Automation) user organization.

We would also point out that the contents of this documentation are not part of a previous or existing agreement, assurance or legal relationship and are not intended to amend the same. All obligations of the manufacturer result from the underlying purchase contract, which also contains the complete and solely valid warranty regulation. These contractual warranty provisions are neither extended nor limited by the production of this documentation.

The manufacturer reserves the right to correct or amend the contents and the product information as well as omissions without prior notification and assumes no kind of liability for damage, injuries or expenditure to be put down to the aforementioned reasons.

These instructions are not to be understood as fundamental information on CANopen. They presuppose underlying knowledge of the CANopen protocol.

1.2 Used Pictograms and Signal Words

The following pictograms and signal words are used in the documentation:

Danger!

Danger refers to an immediate threat. Non-compliance with the precaution described will result in death, serious injury or material damage.

Warning!

Warning refers to a possible threat. Non-compliance with the warning may result in death, serious injury or material damage.

Caution!

Caution refers to an immediate hazard. Non-compliance may result in personal or material damage.

Attention!

Attention and the related text refer to a possible behavior or an undesired condition which can occur during operation.

Note

Marks information that facilitates handling for you and supplements the corresponding part of the documentation.

2 General Safety Instructions and Information on Use

Warning!

The specifications and instructions contained in the documentation must be complied with strictly during installation and commissioning. Before starting the relevant activity, read the documentation carefully and comply with the safety instructions. The term "Qualified Staff" refers to anybody who is familiar with the installation, assembly, commissioning and operation of the frequency inverter and has the proper qualification for the job.

2.1 General Information

Warning!

The DC-link circuit of the frequency inverter is charged during operation, i.e. there is always the risk of contact with high voltage. Frequency inverters are used for driving moving parts and they may become hot at the surface during operation.

Any unauthorized removal of the necessary covers, improper use, wrong installation or operation may result in serious injuries or material damage.

In order to avoid such injuries or damage, only qualified technical staff may carry out the transport, installation, commissioning, setup or maintenance work required. The standards EN 50178, IEC 60364 (Cenelec HD 384 or DIN VDE 0100), IEC 60664-1 (Cenelec HD 625 or VDE 0110-1) as well as the applicable national regulations must be complied with. The term „Qualified Staff" refers to anybody who is familiar with the installation, assembly, commissioning and operation of the frequency inverter as well as the possible hazards and has the proper qualification for the job.

Persons who are not familiar with the operation of the frequency inverter and children must not have access to the device.

2.2 Purpose of the Frequency Inverters

Warning!

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/EEC and EN 60204. In accordance with the CE marking requirements, the frequency inverters comply with the Low Voltage Directive 2006/95/EC as well as EN 61800-5-1. The user shall be responsible for making sure that the requirements of the EMC Directive 2004/108/EEC are met. Frequency inverters are only available at specialized dealers and are exclusively intended for professional use as per EN 61000-3-2.

Purposes other than intended may result in the exclusion of warranty.

The frequency inverters are also marked with the UL label according to UL508c, which proves that they also meet the requirements of the CSA Standard C22.2-No. 14-95.

The technical data, connection specifications and information on ambient conditions are indicated on the name plate and in the documentation and must be complied with in any case. Anyone involved in any kind of work at the device must have read the instructions carefully and understood them before starting the work.

2.3 Transport and Storage

The frequency inverters must be transported and stored in an appropriate way. During transport and storage the devices must remain in their original packaging.

The units may only be stored in dry rooms which are protected against dust and moisture. The units may be exposed to little temperature deviations only. Observe the conditions according to EN 60721-3-1 for storage, EN 60721-3-2 for transport and the marking on the packaging.

The duration of storage without connection to the permissible nominal voltage may not exceed one year.

2.4 Handling and Installation

Warning!

Damaged or destroyed components must not be put into operation because they may be a health hazard.

The frequency inverters are to be used in accordance with the documentation as well as the applicable directives and standards.

They must be handled carefully and protected against mechanical stress.

Do not bend any components or change the isolating distances.

Do not touch electronic components or contacts. The devices are equipped with components which are sensitive to electrostatic energy and can be damaged if handled improperly. Any use of damaged or destroyed components shall be considered as a non-compliance with the applicable standards.

Removal of seal marks may cause restrictions on warranty.

Do not remove any warning signs from the device.

2.5 Electrical Installation

Warning!

Before any assembly or connection work, discharge the frequency inverter. Verify that the frequency inverter is discharged.

Do not touch the terminals because the capacitors may still be charged.

Comply with the information given in the operating instructions and on the frequency inverter label.

Comply with the rules for working on electrical installations.

Rules for working on electrical installation:

- Separate completely (isolate the installation from all possible sources of electrical power.
- Fix (protect against reconnection). Reconnection must be carried out by suitably qualified persons.
- Verify there is no electrical power. Verify that there is no voltage against earth on the plant component by measuring with measurement device or voltage tester.
- Ground and connect in a short circuit. Connect earth conductors.
- Protect against nearby power sources and delimit the working zone.

[1] In plants with a nominal power up to 1 kV deviation from description may be possible.

When working at the frequency inverters, comply with the relevant accident prevention regulations, the applicable standards, standards governing work on systems with dangerous voltages (e.g. EN 50178), directives for electrical and mechanical equipment erection and other national directives.

Comply with the electrical installation instructions given in the documentation as well as the relevant directives.

Responsibility for compliance with and examination of the limit values of the EMC product norm EN 61800-3 for variable-speed electrical drive mechanisms is with the manu-facturer of the industrial plant or machine. The documentation contains information on EMC-conforming installation.

The cables connected to the frequency inverters may not be subjected to high-voltage insulation tests unless appropriate circuitry measures are taken before.

Do not connect any capacitive loads.

2.6 Information on Use

Warning!

The frequency inverter may be connected to power supply every 60 s. This must be considered when operating a mains contactor in jog operation mode. For commissioning or after an emergency stop, a non-recurrent, direct restart is permissible.

After a failure and restoration of the power supply, the motor may start unexpectedly if the auto start function is activated.

If staff is endangered, a restart of the motor must be prevented by means of external circuitry.

Before commissioning and the start of the operation, make sure to fix all covers and check the terminals. Check the additional monitoring and protective devices according to EN 60204 and applicable the safety directives (e.g. Working Machines Act, Accident Prevention Directives etc.).

No connection work may be performed, while the system is in operation.

2.6.1 Using external products

Please note, that Bonfiglioli Vectron does not take any responsibility for the compatibility of external products (e.g. motors, cables, filters, etc.).

To ensure the best system compatibility, Bonfiglioli Vectron offers components which simplify commissioning and provide the best tuning with each other during operation.

Using the device in combination with external products is carried out at your own risk.

2.7 Maintenance and Service

Warning!

Unauthorized opening and improper interventions can lead to personal injury or material damage. Repairs on the frequency inverters may only be carried out by the manu-facturer or persons authorized by the manufacturer.

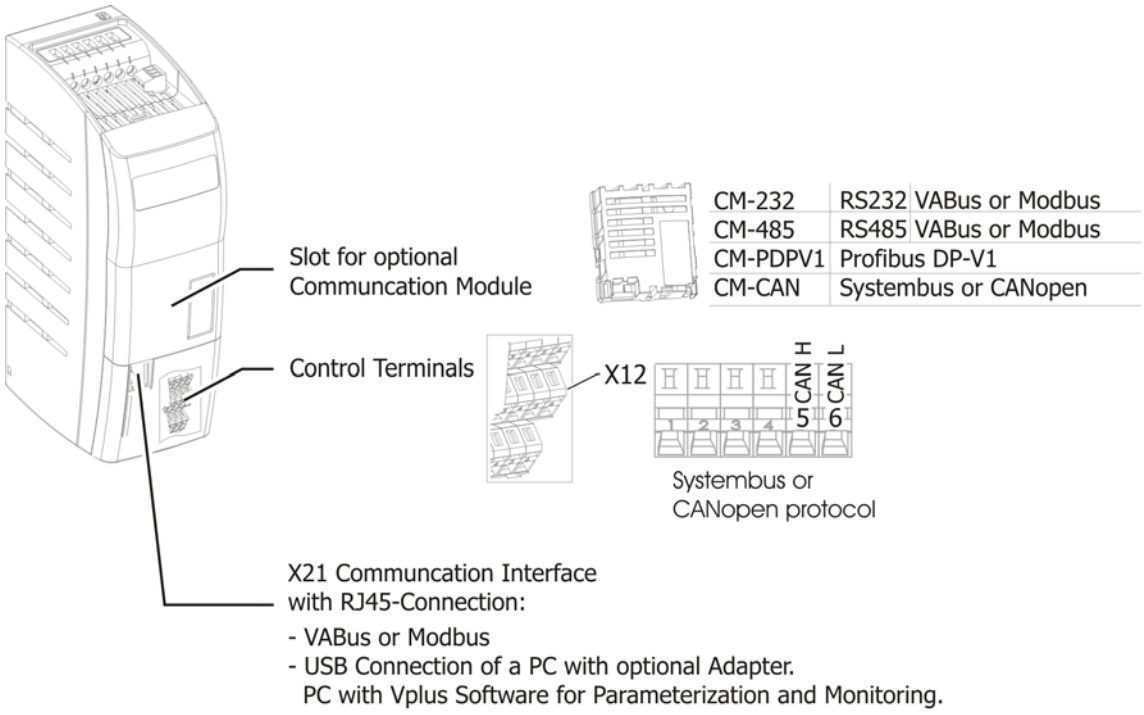
Check protective equipment regularly.

Any repair work must be carried out by qualified electricians.

2.8 Disposal

The dispose of frequency inverter components must be carried out in accordance with the local and country-specific regulations and standards.

3 Communication Option



Interface	See
Control Terminals for CAN-Connection	Instructions for Systembus or CANopen®[2].
CM-CAN	
X21 Communication Interface[3]	Instructions for VABus or Modbus.
CM-232	Instructions for VABus or Modbus.
CM-485	Instructions for VABus or Modbus.
CM-PDPV1	Instructions for Profibus DP-V1.

Combinations of Systembus and CANopen® communication on the two interfaces:

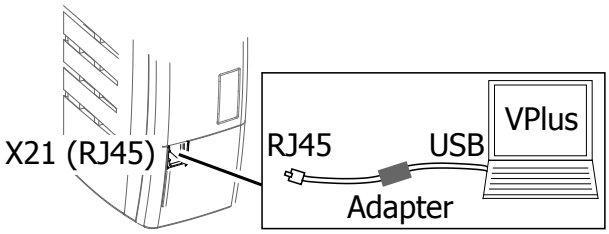
Optional Communication Module (CM)		Frequency Inverter Terminals X12.5 and X12.6
CANopen®	as well as	Systembus
Systembus	as well as	CANopen®

[2] The CANopen®-Communication products fulfill the specifications of the CiA® (CAN in Automation) user organization.

[3] Install an interface adapter for connection to a PC. This enables parameterization und monitoring via the VPlus PC-Software.

3.1 VPlus PC-Software

The USB-Interface of a PC can be connected to the X21 Communication Interface via an optional USB adapter. This enables parameterization and monitoring with the help of the VPlus PC-Software.



4 Installation of an optional Communication Module

4.1 Assembly

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

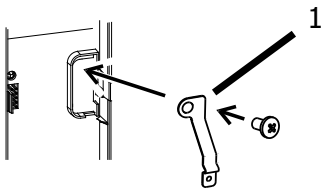
Caution!

The frequency inverter must be disconnected from the power supply before installation of the communication module.

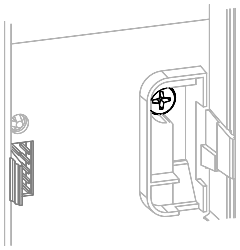
Installation under voltage is not permitted and will destroy the frequency inverter and/or the communication module.

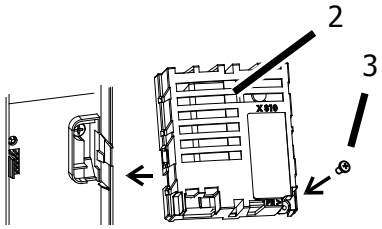
Do not touch the PCB visible on the back of the module, otherwise components may be damaged.

- Remove the cover of the module slot.



- Attach the PE spring (1) using the screw provided on the frequency inverter.

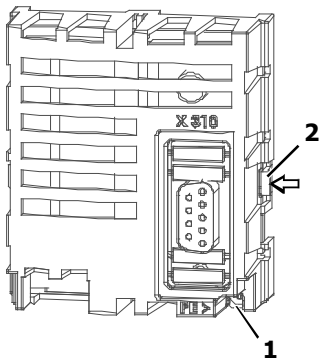




- Insert the Communication Module (2).
- Screw the communication module (2) onto the frequency with the screw (3).
- Break off the pre-punched cut-out from the cover.
- Replace the cover.

4.2 Disassembly

- Remove the cover of the module slot.



- Loosen the screw (1) on the communication module.
- Using a small screwdriver, firstly unlock the right and then the left snap-in hook (2).
- Remove the communication module from the slot.
- Unscrew the PE spring.
- Replace the cover onto the frequency inverter.

5 Connection

The CAN-Connection of the Systembus is physically laid out according to ISO-DIS 11898 (CAN High Speed).

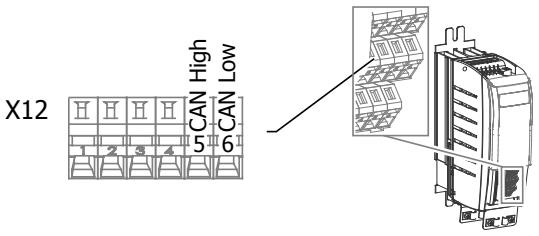
Caution!

The frequency inverter must be disconnected from the power supply before installation and connection work. Make sure that the frequency inverter is de-energized.

Installation under voltage is not permitted and can lead to the destruction of the frequency inverter and/or the communication module.

5.1 Connection to the Terminals

Connect the bus to the X12.5 and X12.6 terminals of the frequency inverter.

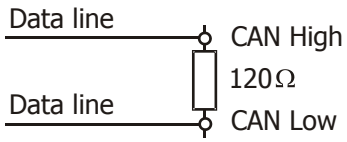


Cable

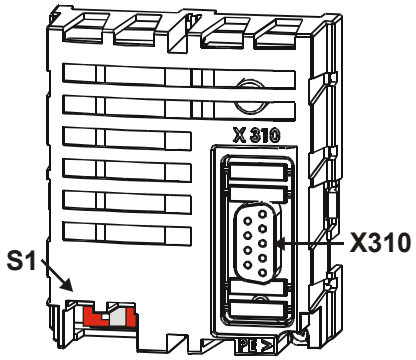
- For the bus line use a twisted and shielded cable.
- Implement the shield as a braided shield (not a foil shield).
- Connect the cable shield surfaces to PE at both ends.

Bus Termination

Connect the bus termination, necessary on one cable, at the first and last physical participant.



5.2 Module Connection

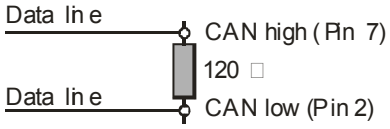


The **X310** (9-pin D-Sub) bus plug has been designed according to DS102 Version 2.0 (Bus node, option A).
Details can be seen from the following table on the occupancy of the bus plug.

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

The factory setting for the bus termination is OFF.

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



Attention!
Make absolutely sure that only one of the two possibilities for the bus termination is used and the bus termination is only switched on with the first and last subscriber. Otherwise, operation of the CANopen® communication is not possible.

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

For the bus line, use twisted cable with harness shield (**no foil shield**).

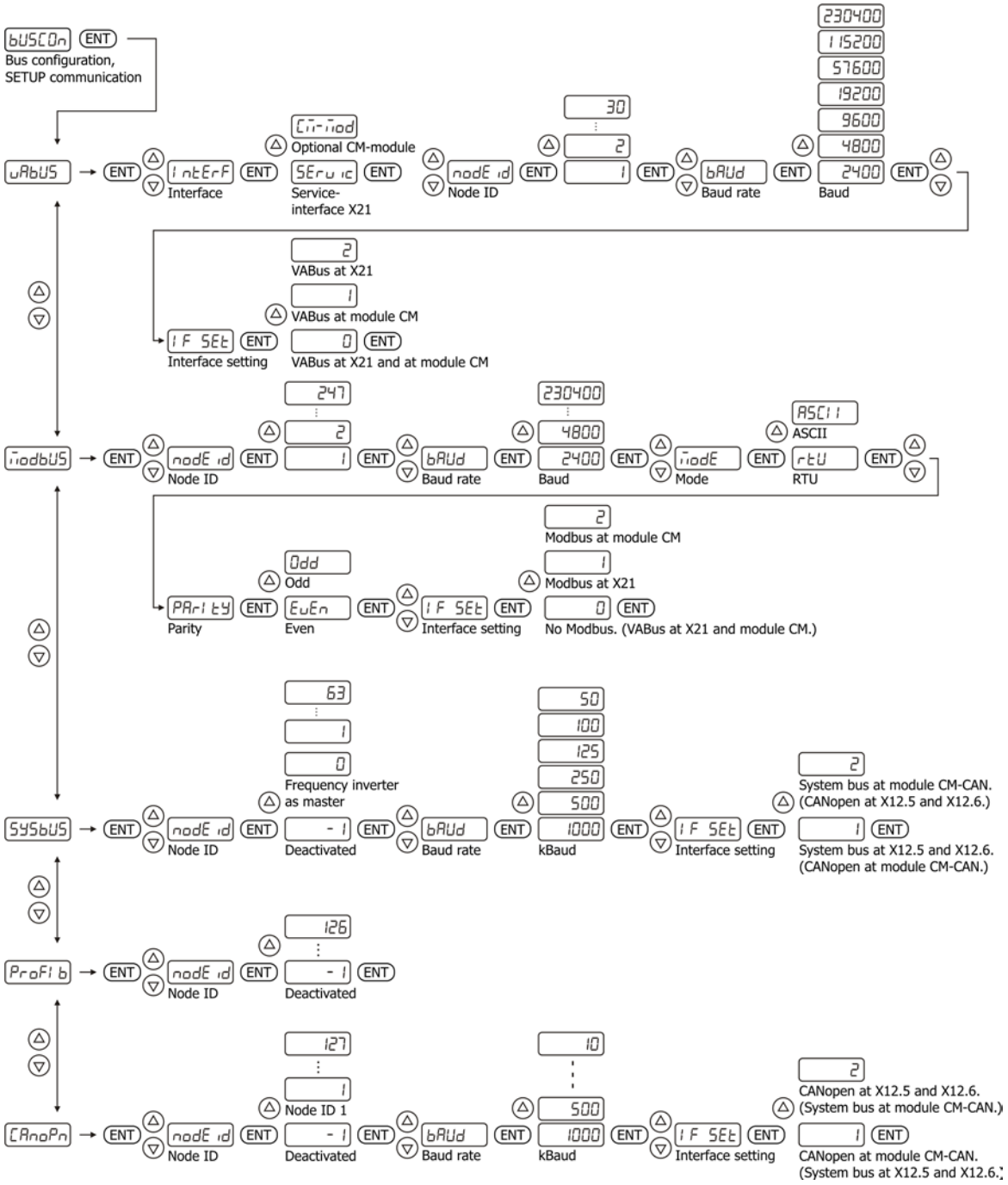
Attention!
The harness screen of the data lines is to be connected to ground (PE) on both sides on a large area.

6 Commissioning via the Operator Panel

A communication interface can be set up in the "Setup" menu of the operator panel. Further communication parameters can be set in the "Para" menu.

6.1 Menu for setting up the Communication

The communication interface can be set up quickly and simple via the Operator Panel.



6.2 Select the Protocol

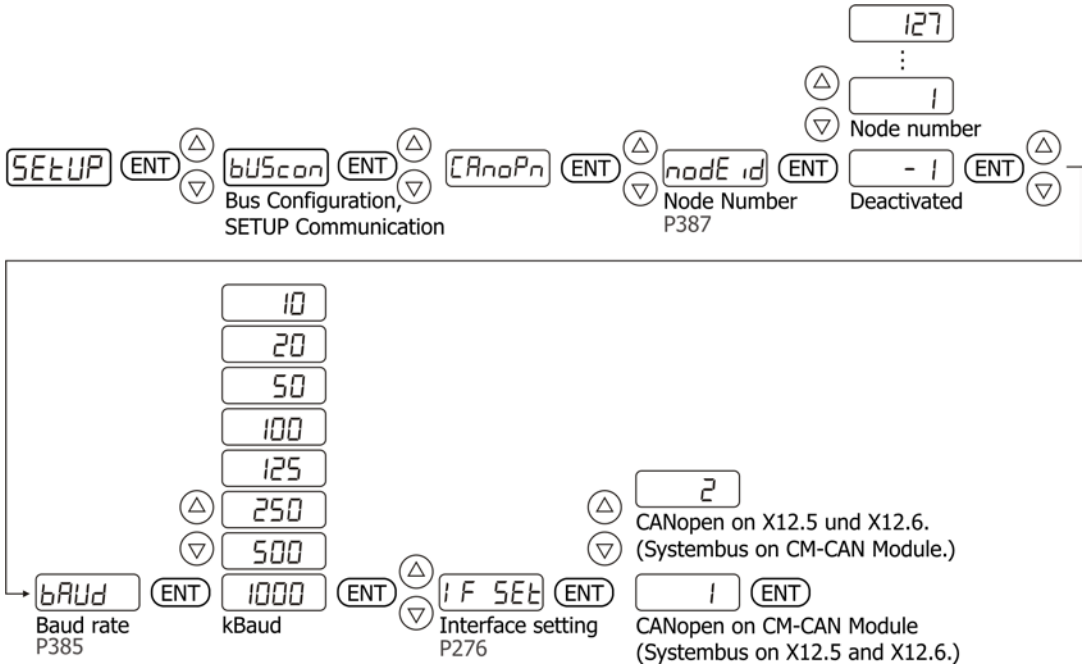
- Select CANopen.

	Display
Select the "Setup" menu using the arrow keys.	SEtUP ENT
Using the arrow keys select:	▲ ▼
Setting up a Communication Interface (Bus Configuration) asd	buSCOn ENT
Select a protocol using the arrow keys:	▲ ▼
CANopen	CAnoPn
Profibus[4]	PrOFI b
Systembus	SYSbUS
Modbus	ModbUS
VABus	uAbUS
	ENT

6.3 Set the Communication Parameters

Parameter		Display
387	Node Number	nodE id
385	Baud Rate	bAUd
276	CAN Interface (CM-CAN/X12). Interface setting.	I F SEt
	– Set the X12.5 and X12.6 Terminals to the CANopen Protocol Or:	2
	– Set an optional CM-CAN Communication Module to CANopen.	1

[4] The selection is only possible if an optional CM-PDPV1 Communication Module is installed.



6.4 Set the Protocol for the Terminals and Communication Module

- **276 CAN Interface (CM-CAN/X12)**

With Parameter *CAN Interface (CM-CAN/X12)* **276** the terminals of the frequency inverter and an optional Communication Module (CM) can be set to a communication protocol. In the factory setting (1 - CANopen/CAN-Systembus) the terminals of the frequency inverter are set to CAN-Systembus.

CM-CAN/CAN-Terminals	Function	
1 - CANopen/CAN-Systembus	Optional Communication Module: Frequency Inverter Terminals X12.5 and X12.6:	CANopen® Systembus
2 - CAN-Systembus/CANopen	Optional Communication Module: Frequency Inverter Terminals X12.5 und X12.6:	Systembus CANopen®

CM-CAN: Optional Communication Module
CAN-Terminals: Terminals X12.5 and X12.6 of the frequency inverter.

Simultaneous CANopen®-Communication via the Terminals of the frequency inverter and via a Communication Module is not possible.

Simultaneous Systembus-Communication via the Terminals of the frequency inverter and via a Communication Module is not possible.

7 CANopen

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 available objects are subdivided into:

Communication Objects	(0x1nnn)	according to DS301 V4.02
Manufacturer Objects	(0x2nnn)	
Standardized Objects	(0x6nnn)	according to DS402 V2.0

The functions and objects are described here as far as is necessary. For further information refer to the CiA Draft Standards.

The standards, referred to here, are DS102, DS301 und DS402. These can be obtained from:

CAN in AUTOMATION (CiA)
Kontumazgarten 3
D-90429 Nürnberg

Tel.: +49 911 928819 0
Fax: +49 911 928819 79
URL: www.can-cia.org
E-Mail: headquarters@can-cia.org

Attention!
With the help of the CM-CAN CANopen® communication module, it is possible to access **ALL** parameters of the frequency inverter from the external control unit. Control of the access via the operation level, as with the Operator Panel or the VPlus PC-Software, does not exist. A change of parameters with an unknown meaning to the user can lead to the inoperability of the frequency inverter.

Attention!
If data is written cyclically comply with the instructions in Chapter 9.3.1 "Handling of Data Sets / Cyclic Writing".

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

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

7.1 Baud Rate Setting / Line Lengths

- **385 CAN Baud rate**

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

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

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

CANopen Interface		
Operation Mode	**Function**	**max. Line Length**
1 - 10 kBaud	Transmission rate 10 kBaud	5000 Meter
2 - 20 kBaud	Transmission rate 20 kBaud	2500 Meter
3 - 50 kBaud	Transmission rate 50 kBaud	1000 Meter
4 - 100 kBaud	Transmission rate 100 kBaud	500 Meter
5 - 125 kBaud	Transmission rate 125 kBaud	500 Meter
6 - 250 kBaud	Transmission rate 250 kBaud	250 Meter
7 - 500 kBaud	Transmission rate 500 kBaud	100 Meter
8 - 1000 kBaud	Transmission rate 1000 kBaud	25 Meter

7.2 Setting the Node Address

- **387 CAN Node-Number**

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

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Factory setting**
387	CAN Node Number	-1	127	-1

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

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

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

7.3 Operational Behavior on Bus Failure

- **388 CAN Error Behavior**

The operational behavior if the CANopen® system fails due to BusOff, guarding, heartbeat, SYNC error, RxPDO length error or NMT state change (leaving NMT state operational) can be parameterized. The required behavior is set with parameter *CAN Error Behavior* **388**.

CAN Error Behavior 388	Function
0 - No Reaction	Operating point is maintained
1 - Error	Immediate change of state to "Fault". Factory Setting.
2 - Switch-off	Control command "Disable Voltage" and change of state to "Voltage disabled".
3 - Quick-Stop	Control command "Quick Stop" and change of state to "Switch-on disabled".
4 - Ramp-Stop + Error	Control command "Disable Operation" and change of state to "Fault", after the drive has been stopped.
5 - Quick-Stop + Error	Control command "Quick Stop" and change of state to "Fault", after the drive has been stopped.

Attention!
The parameter settings *CAN Error Behavior* **388** = 2 ... 5 are only relevant if parameter *Local/Remote* **412** = "1 - Control via Statemachine" has been set.

Parameter *CAN Error Behavior* **388** corresponds to the device profile object 0x6007 *Abort Connection option code*.

The functional behavior of the frequency inverter is described in detail in Chapter 9.5.1 "0x6007/0 Abort Connection option code".

The error and warning behavior of the frequency inverter can be parameterized in various ways. If a failure of the bus system occurs in the setting *CAN Error Behavior* **388** = 1, 4 or 5, the frequency inverter reports one of the following errors:

Communication Errors		
Code		**Meaning**
F20	21	Bus OFF
	22	Guarding failure
	23	Error state
	24	SYNC error (SYNC timing)
	25	NMT state change (operational → xxx)
	26	RxPDO1 length error (number of received bytes different to mapping)
	27	RxPDO2 length error (number of received bytes different to mapping)
	28	RxPDO3 length error (number of received bytes different to mapping)
F23	nn	Heartbeat failure – nn = Node Address of the failed subscriber (hex)

8 Protocol

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

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

8.1 Communication Objects

The communication objects are located in the index range 0x1nnn. They describe the communication behavior 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.

8.2 Application Objects

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

8.3 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.

The necessary messages for reading and writing objects with "expedited transfer" are described in detail in Chapter 9.3 "Manufacturer Objects (0x2nnn) – Parameter Access". 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 frequency inverter supports one Server SDO. This Server SDO is accessed by the Client SDO on the PLC side. An SDO message always consists of a COB-ID followed by 8 data bytes.

SDO-Message:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Command specifier (cs)	Index		Sub-Index	Data	Data	Data	Data
	nn	LSB	MSB					

Default Identifiers (COB-ID):

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

Depending on the transfer direction and the amount of data bytes, different command specifiers are used.

The error codes are listed in chapter 9.5.2 "0x603F/0 Error Code".

8.3.1 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 number of valid data bytes is coded in the command specifier of the response.

No. of Data Bytes	**1**	**2**	**3**	**4**
Command specifier (cs)	0x4F	0x4B	0x47	0x43

8.3.2 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		00	00	00	00

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		data01	data02	data03	data04

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

No. of Data Bytes	**1**	**2**	**3**	**4**
Command specifier	0x2F	0x2B	0x27	0x23

8.3.3 Error Code Table

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

Server → Client Abort SDO Transfer

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

Error Codes			
Abort-code high	**Abort-code low**	**CANopen Description**	**Product-Specific Allocation**
0x0601	0x0000	Unsupported access to an object	Parameter cannot be written or read
0x0602	0x0000	Object does not exist	Parameter does not exist.
0x0604	0x0047	General internal incompatibility in the device	Data Sets differ
0x0606	0x0000	Access failed due to a hardware error	EEPROM Error (Read/ Write/ Checksum)
0x0607	0x0010	Data Type does not match	Parameter has a different data type.
0x0607	0x0012	Data type does not match or length of Service telegram too big	Parameter has a different data type or telegram length not correct.
0x0607	0x0013	Data type does not match or length of Service telegram too small	Parameter has a different data type or telegram length not correct.
0x0609	0x0011	Sub-Index does not exist	Data Set does not exist.
0x0609	0x0030	Value range of parameter exceeded	Parameter Value too large or too small.
0x0609	0x0031	Value of parameter written too high.	Parameter Value too large.
0x0609	0x0032	Value of parameter written too low.	Parameter Value too small.
0x0800	0x0020	Data cannot be transmitted or saved	Invalid value for operation.
0x0800	0x0021	Data cannot be transferred because of local control	Parameter cannot be written during operation.

8.4 PDO Function

The PDO (Process Data Objects) messages are messages with up to eight bytes of process data. The process data objects are mapped to the Rx/Tx-PDOs with the help of communication objects (communication/mapping parameter). The frequency inverters support 3 RxPDOs (PLC ➔ inverter) and 3 TxPDOs (inverter ➔ PLC).

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

PDO-Message:

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

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

Byte	0	1	2	3	4	5
	16 Bit-Object		32 Bit-Object			
	LSB	MSB	LSB	…	…	MSB

Predefined Identifiers (COB-ID):

	Decimal	Hexadecimal
TxPDO1	384 + Node-ID	180 + Node-ID
RxPDO1	512 + Node-ID	200 + Node-ID
TxPDO2	640 + Node-ID	280 + Node-ID
RxPDO2	798 + Node-ID	300 + Node-ID
TxPDO3	896 + Node-ID	380 + Node-ID
RxPDO3	1024 + Node-ID	400 + Node-ID

Node-ID = Node number

8.5 Emergency Function

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

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

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

The Manufacturer Error Code corresponds to the inverter error codes described in the Operating Instructions and in this documentation in Chapter 12.5 "Error Messages".

8.6 SYNC

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 8.10 "OS Synchronization".

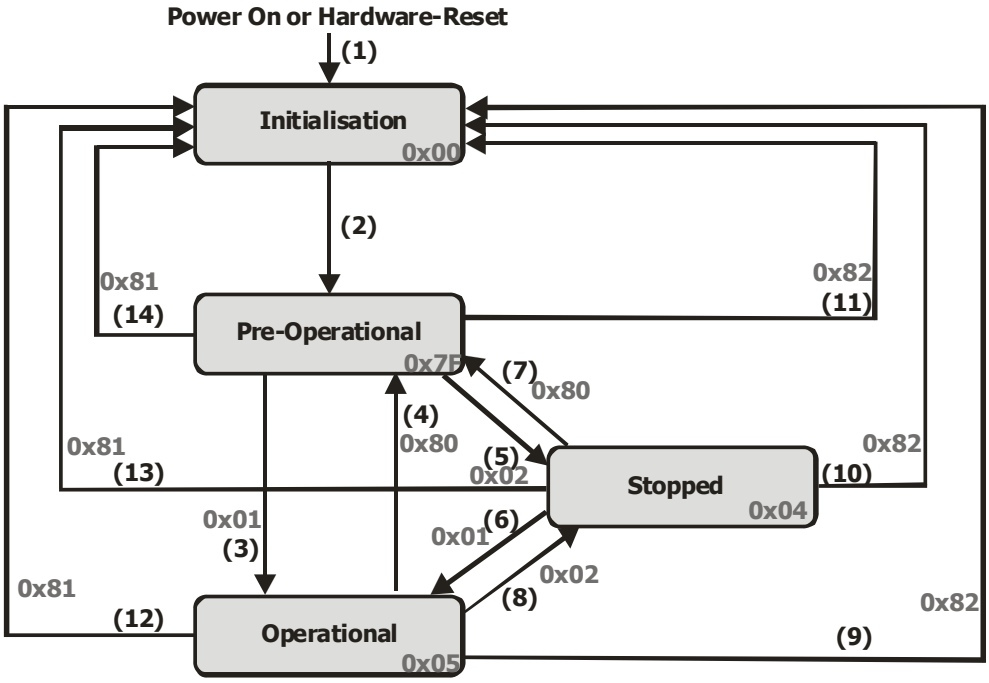
The SYNC message is a message with no data or with one byte data which is ignored. The default Identifier = 0x80 (=128).

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

8.7 NMT Functions

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

8.7.1 NMT Statemachine



Note:
A change of NMT-State may also be triggered by a communication (Bus-off, Guarding, etc.). The behavior of the NMT state machine in such a case is described in Chapter 9.2.17 "0x1029/n Error Behavior".

Transition	NMT Command
(1)	At power on NMT state Initialization is entered autonomously.
(2)	NMT state Initialization 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 Object 0x1nnn and Application Objects 0x6nnn are reset.
(12), (13), (14)	Reset Communication. Communication Objects 0x1nnn are reset.

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

8.7.2 Boot-Up Message

Identifier	Byte
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 Node guarding.

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 or node communication and results in a Boot-Up message.

8.7.3 NMT Commands

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

id = 0 Command accepted by all devices
id = 1…0x7F =(127) Command accepted by 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

8.8 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:
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 behavior:
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 *0x6007/0 Abort Connection option code* and *0x1029/n Error Behavior*.

Guarding process:
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

Frequency inverter:

	Byte 0							
Identifier	NMT State + Toggle Bit							
0x700 + Node-ID	7	6	5	4	3	2	1	0
	t	NMT State						

t: The toggle bit is toggled on each transmission (first transmission: t = 0)

NMT State:

0	Boot-Up
4	Stopped
5	Operational
0x7F (=127)	Pre-Operational

8.9 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 *0x6007/0 Abort Connection option code* and *0x1029/n Error Behavior*.

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

Heartbeat Message

	Byte 0							
Identifier	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

8.10 OS Synchronization

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

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

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

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

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

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

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

The CANopen "active status for synchronization" is recognized by the parameter setting **387** *CAN Node Number* >0 and a running synchronous PDO.

The Systembus "active status for synchronization" is recognized by the parameter setting **900** *Systembus Node ID* >0. Also parameter **1180** *Synchronization* has to be set to SYNC or an RxPDO.

- **1180 Operation Mode (Synchronization)**

Operation Mode 1180	Function
0 - Aus	No evaluation of SYNC-Telegrams.
1 - RxPDO1	RxPDO1 used as SYNC-Telegram.
2 - RxPDO2	RxPDO2 used as SYNC-Telegram.
3 - RxPDO3	RxPDO3 used as SYNC-Telegram.
10 - SYNC	SYNC used as SYNC-Telegram.

With Parameter *Operation Mode* **1180** the telegram for the synchronisation on the Systembus is determined.

The selected telegram is used

- for the synchronisation of the operating system (if Parameter *OS_SyncSource* **1452** is set to "Systembus") and
- for the processing of the synchronous PDOs on the Systembus (if set in Parameters 930, 932, 934, 936-938).

- **1451 CANopen OS Synctime**

Parameter *CANopen OS Synctime* **1451** 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.

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

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

For the CANopen Synchronisation pay also attention to Objects *0x1005/0 COB-ID SYNC Message*, *0x1006/0 Communication Cycle Period* and *0x1007/0 Synchronous Window Length*.

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

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

9 Objects

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

9.1 Tabular Object Overview

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

Access Type			
Read only	The PLC is only allowed to read the data from the frequency inverter.		
Read/Write	The PLC is granted full access (read and write) 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)

PDO Mapping	
No	This object cannot be used for PDO exchange, only SDO is applicable.
Tx	This object can be transmitted in a TxPDO from the frequency inverter.
Rx	This object can be transmitted in a RxPDO to the frequency inverter.

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

9.1.1 Communication Objects

Index	Sub-Index	Name	SDO Access	Data Type	PDO-Mapping
0x1000	0	Device type	Read only	Unsigned32	No
0x1001	0	Error register	Read only	Unsigned8	No
0x1005	0	COB-ID SYNC object	Read/Write	Unsigned32	No
0x1006	0	Communication cycle period	Read/Write	Unsigned32	No
0x1007	0	Synchronous window length	Read/Write	Unsigned32	No
0x1008	0	Manufacturer device name	Read only	Visible string	No
0x1009	0	Manufacturer hardware version	Read only	Visible string	No
0x100A	0	Manufacturer software version	Read only	Visible string	No
0x100C	0	Guard time	Read/Write	Unsigned16	No
0x100D	0	Life time factor	Read/Write	Unsigned8	No
0x1010	0	Store parameters Highest Sub-Index supported	Read only	Unsigned8	No
	1	Save all parameters	Read/Write	Unsigned32	No
	2	Save communication parameters	Read/Write	Unsigned32	No
	3	Save application parameters	Read/Write	Unsigned32	No
0x1011	0	Restore default parameters Highest Sub-Index supported	Read only	Unsigned8	No
	1	Restore all default parameters	Read/Write	Unsigned32	No
	2	Restore communication default parameters	Read/Write	Unsigned32	No
	3	Restore application default parameters	Read/Write	Unsigned32	No

Index	Sub-Index	Name	SDO Access	Data Type	PDO-Mapping
0x1014	0	COB-ID emergency object	Read/Write	Unsigned32	No
0x1016	0	Consumer heartbeat time Highest Sub-Index supported	Read only	Unsigned8	No
	1	Consumer heartbeat time 1	Read/Write	Unsigned32	No
	2	Consumer heartbeat time 2	Read/Write	Unsigned32	No
	3	Consumer heartbeat time 3	Read/Write	Unsigned32	No
0x1017	0	Producer heartbeat time	Read/Write	Unsigned16	No
0x1018	0	Identity object Highest Sub-Index supported	Read only	Unsigned8	No
	1	Vendor ID	Read only	Unsigned32	No
	2	Product code	Read only	Unsigned32	No
	3	Revision number	Read only	Unsigned32	No
	4	Serial number	Read only	Unsigned32	No
0x1029	0	Error behavior	Read only	Unsigned8	No
	1	Communication error	Read/Write	Unsigned8	No
0x1200	0	Server SDO parameter	Read only	Unsigned8	
	1	COB-ID Rx	Read only	Unsigned32	No
	2	COB-ID Tx	Read only	Unsigned32	No
0x1400	0	RxPDO1 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1401	0	RxPDO2 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1402	0	RxPDO3 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1600	0	RxPDO1 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No
0x1601	0	RxPDO2 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No

Index	Sub-Index	Name	SDO Access	Data Type	PDO-Mapping
0x1602	0	RxPDO3 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No
0x1800	0	TxPDO1 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time		Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1801	0	TxPDO2 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time	Read/Write	Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1802	0	TxPDO3 communication parameter Highest Sub-Index supported	Read only	Unsigned8	No
	1	COB-ID	Read/Write	Unsigned32	No
	2	Transmission type	Read/Write	Unsigned8	No
	3	Inhibit time	Read/Write	Unsigned16	No
	4	-	-	-	No
	5	Event time	Read/Write	Unsigned16	No
0x1A00	0	TxPDO1 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No
0x1A01	0	TxPDO2 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No
0x1A02	0	TxPDO3 mapping parameter No. of mapped objects	Read/Write	Unsigned8	No
	1	1. mapped obj.	Read/Write	Unsigned32	No
	2	2. mapped obj.	Read/Write	Unsigned32	No
	3	3. mapped obj.	Read/Write	Unsigned32	No
	4	4. mapped obj.	Read/Write	Unsigned32	No
	5	5. mapped obj.	Read/Write	Unsigned32	No
	6	6. mapped obj.	Read/Write	Unsigned32	No
	7	7. mapped obj.	Read/Write	Unsigned32	No
	8	8. mapped obj.	Read/Write	Unsigned32	No

9.1.2 Manufacturer Objects

Index	Sub-Index	Name	SDO Access	Data Type	PDO-Mapping
0x2nnn	0, 1, … 9	Manufacturer specific Direct access to inverter parameters Read/write access by SDO transfer only			
0x3000	0	Sync Jitter	Read/Write	Unsigned16	No
0x3001	0	Digital In actual values	Read only	Unsigned16	Tx
0x3002	0	Digital Out actual values	Read only	Unsigned16	Tx
0x3003	0	Digital Out set values	Read/Write	Unsigned16	Rx
0x3004	0	Boolean Mux	Read only	Unsigned16	Tx
0x3005	0	Boolean Demux	Read/Write	Unsigned16	Rx
0x3006	0	Percentage set value	Read/Write	Unsigned16	Rx
x3007	0	Percentage actual value	Read only	Unsigned16	Tx

9.1.3 Device Profile Objects

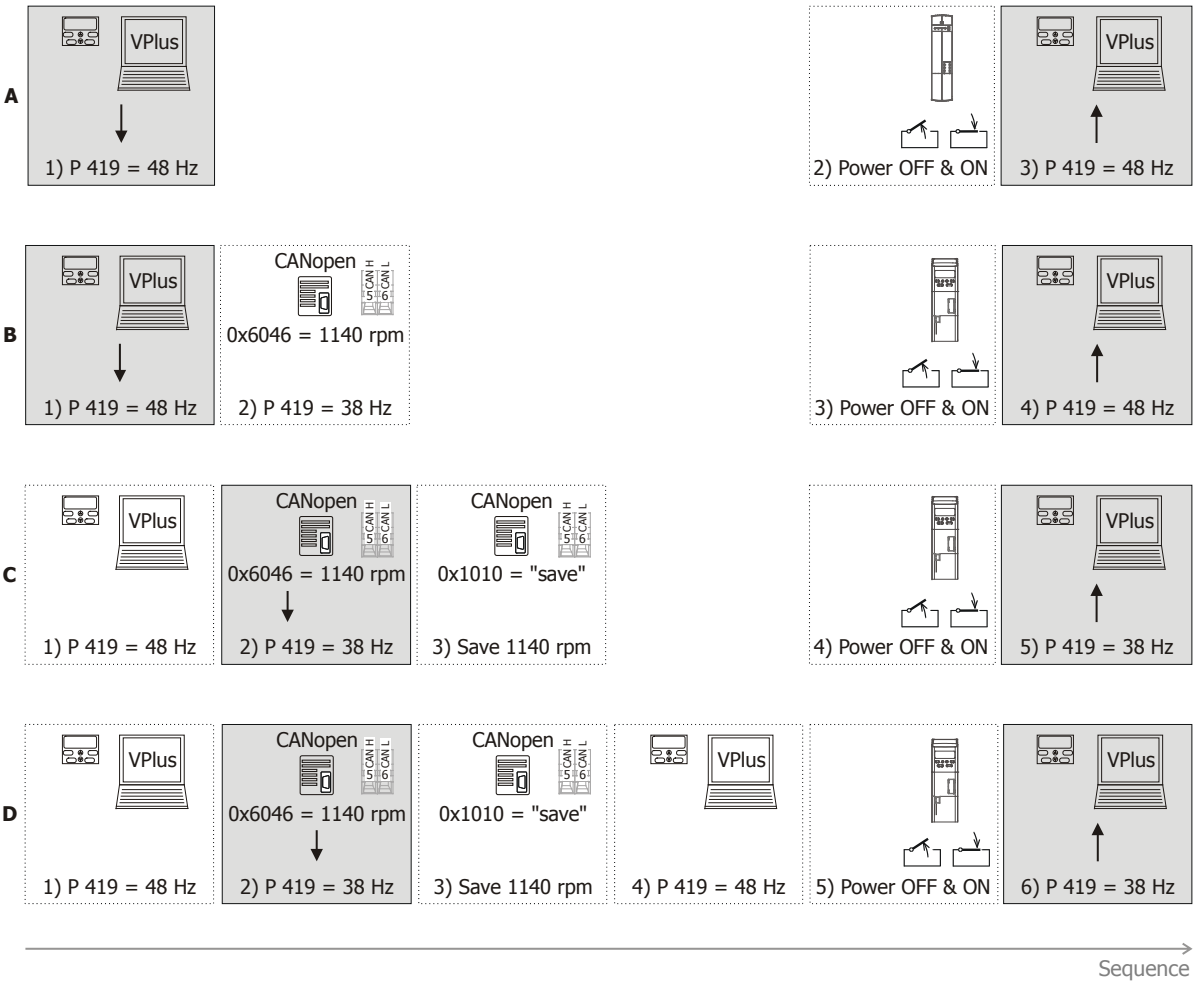
Index	Sub-Index	Name	SDO Access	Data Type	PDO-mapping	Factory setting	Min...Max	Corresp. Param.
0x6007	0	Abort connection option code	Read/write	Integer16	No	1	-2…3	p.388
0x603F	0	Error code	Read only	Unsigned16	No	-	-	-
0x6040	0	controlword	Read/write	Unsigned16	Rx	-	-	p.410
0x6041	0	statusword	Read/only	Unsigned16	Tx	-	-	p.411
0x6042	0	Target velocity	Read/write	Interger16	Rx	0	-32768… 32767	-
0x6043	0	Target velocity de-mand	Read only	Integer16	Tx	-	-	-
0x6044	0	Control effort	Read only	Integer16	Tx	-	-	-
0x6046		Velocity min max						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Velocity min amount	Read/write	Unsigned32	No	0	0…32767	p.418
	2	Velocity max amount	Read/write	Unsigned32	No	32767	0…32767	p.419
0x6048		Velocity acceleration				-	-	-
	0	Highest sub-index supported	Read only	Unsigned8	No			
	1	Delta speed	Read/write	Unsigned32	No	150	1…32767	p.420
	2	Delta time	Read/write	Unsigned16	No	1	1…65535	p.422
0x6049		Velocity deceleration						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Delta speed	Read/write	Unsigned32	No	150	1…32767	p.421
	2	Delta time	Read/write	Unsigned16	No	1	1…65535	p.423
0x604A		Velocity quick stop						
	0	Highest sub-index supported	Read only	Unsigned8	No	-	-	-
	1	Delta speed	Read/write	Unsigned32	No	150	1…32767	p.421
	2	Delta time	Read/write	Unsigned16	No	1	1…65535	p.423
0x6060	0	Modes of operation	Write only	Integer8	Rx	2	2	-
0x6061	0	Modes of operation display	Read only	Integer8	Tx	2	-	-
0x6071	0	Target torque	Read/write	Integer16	Rx			-
0x6077	0	Torque actual value	Read only	Integer16	Tx			p.224
0x6078	0	Current actual value	Read only	Integer16	Tx			p.214
0x6079	0	DClink circuit voltage	Read only	Integer32	Tx			p.222

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

Attention!
Some of the above listed CANopen® objects have corresponding inverter parameters.
These objects are handled in a special way. If one of these CANopen® objects has been written by SDO followed by a "save" command (see Object *0x1010*), the written value is stored to non-volatile memory of the inverter. After the next power on of the inverter these CANopen® object values are restored again and overwrite the inverter parameter values..

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

Effect of the "Save" command (Object *0x1010*)
(Examples for the sequence of writing parameters and objects)



A Value of a parameter is set via the Operator Panel or VPlus. No "Save" command.

1) Setting of *Maximum Frequency* **419** = 48 Hz on Operator Panel or in VPlus.
2) Power OFF and ON.
3) The value of the Operator Panel/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 on Operator Panel or in VPlus.
2) Setting of CANopen® object *0x6046* = 1140 rpm* (equivalent to 38 Hz).
3) Power OFF and ON.
4) The value of the CANopen® object is overwritten with the parameter value of the Operator Panel or VPlus. The value of the Operator Panel or from VPlus is active (48 Hz).

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

1) Setting of *Maximum Frequency* **419** = 48 Hz on Operator Panel 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 on Operator Panel 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 on Operator Panel or in VPlus.
5) Power OFF and ON.
6) The parameter value is overwritten with the value of the CANopen® object *0x6046*. The value of the 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).

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

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

9.2 Communication Objects (0x1nnn)

The communication objects 0x1nnn contain all parameters for the communication.
Note: For easier use, the objects are summarized by a table in each paragraph. This table is additionally marked by color.

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

Used Abbreviations:

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

Map: Mapping
Def.-Val: Default value

The Examples show some typical data telegrams, which could be observed or used with a CAN analyzing tool. The order displayed in the examples is the standard CANopen® format, lowest byte left, highest byte right.

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

9.2.1 0x1000/0 Device Type

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

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

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

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

Device Profile Number	= 402	drives and motion control
Type	= 1	frequency converter
Type	= 2	servo drive
Mode bits	= 0	unused

Example:

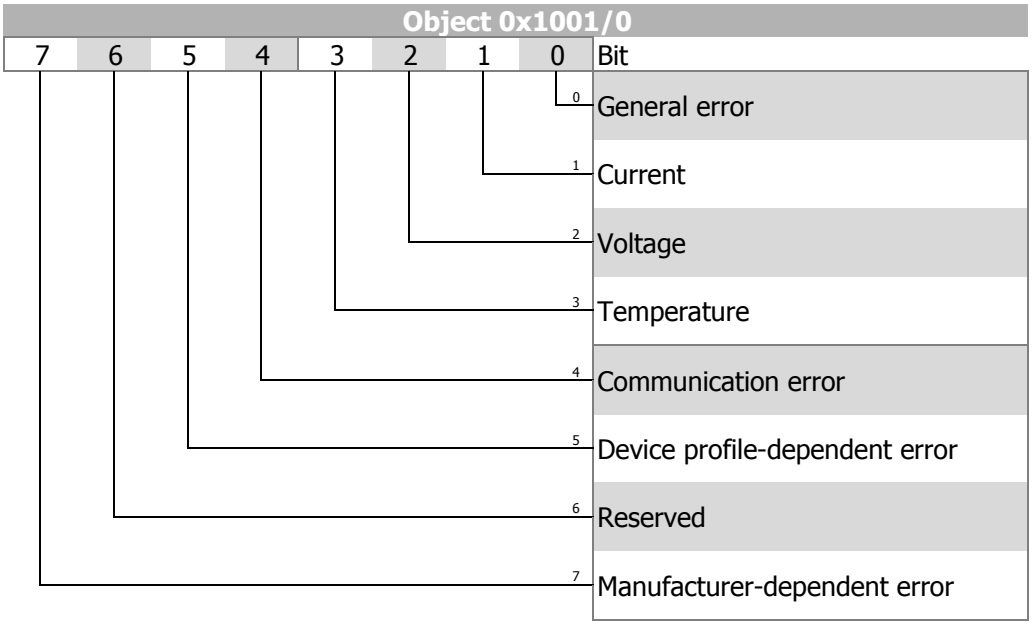
	COB ID	CB	Index	SI	Data
Read Request	601	40	00 10	00	00 00 00 00
Response	581	43	00 10	00	92 01 41 00

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

9.2.2 0x1001/0 Error Register

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

Object 0x1001/0 is the error register for internal errors of the frequency inverter. The status error-free (0x1001/0 = 0) or error exists (0x1001/0 ≠ 0) is displayed.



Example:

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

CB: Control byte SI: Sub-Index

9.2.3 0x1005/0 COB-ID SYNC Message

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

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

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

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

Bit 31: X = don't care

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

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

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

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

CB: Control byte SI: Sub-Index

9.2.4 0x1006/0 Communication Cycle Period

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

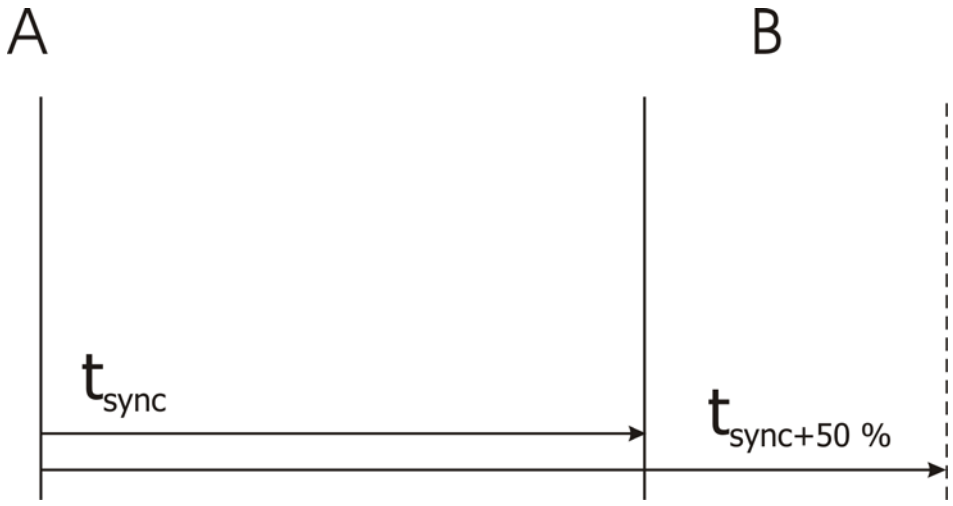
The *communication cycle period* is the time distance between two consecutive SYNC messages.
The value for *communication cycle period* is given in multiples of micro seconds.
Values smaller than 20000 (20 ms) are allowed.

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*.

Note:
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.



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

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

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

Example:

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

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

9.2.5 0x1007/0 Synchronous Window Length

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

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

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

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

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

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

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

9.2.6 0x1008/0 Manufacturer Device Name

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

The device version is displayed as a sequence of ASCII characters.
Example: "**AGILE**"

Object 0x1008/0 supports the segmented SDO transfer.

9.2.7 0x1009/0 Manufacturer Hardware Version

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

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

Object 0x1009/0 supports the segmented SDO transfer.

9.2.8 0x100A/0 Manufacturer Software Version

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

The software version is displayed as a sequence of ASCII characters.
Example: "**5.1.2**"

Object 0x100A/0 supports the segmented SDO transfer.

9.2.9 0x100C/0 Guard Time

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

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

Example:

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

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

9.2.10 0x100D/0 Lifetime Factor

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

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

Example:

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

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

9.2.11 0x1010/n Store Parameters

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

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

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

Specification for writing the "save" command

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

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

Example:

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

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

9.2.12 0x1011/n Restore default Parameters

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

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

Writing "load" to 0x1011/3 restores all application parameters (*0x6nnn*).

Specification for writing the "load" command

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

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

Example:

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

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

9.2.13 0x1014/0 COB-ID Emergency Message

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

The identifier and thus the definition of the priority for the emergency message can be set with object 0x1014/0. The default value of the identifier is 128 + Node ID (valid)

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

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

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

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

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

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

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

Example:

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

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

9.2.14 0x1016/n Consumer Heartbeat Time

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

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

Value of *Consumer Heartbeat Time*		
Bit 24 to Bit 31	Bit 16 to Bit 23	Bits 0 to Bit 15
not used	Node ID	Heartbeat Time

Example:

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

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

9.2.15 0x1017/0 Producer Heartbeat Time

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

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

Example:

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

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

9.2.16 0x1018/n Identity Object

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

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

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

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

Example:

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

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

9.2.17 0x1029/n Error Behavior

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

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

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

Example:

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

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

9.2.18 0x1200/n SDO Server Parameter

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

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

COB-ID client → server (Rx) = 1536 + Node Address

COB-ID server → client (Tx) = 1408 + Node Address

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

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

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

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

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	00 12	02	00 00 00 00
Response	581	43	00 12	02	01 06 00 00

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

9.2.19 0x1400/n, 0x1401/n, 0x1402/n RxPDO Communication Parameter

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 exists / valid
1 = PDO does not exist / 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 RxPDOs. If during the set time no RxPDO is received, one of the following faults is triggered:
202A RxPDO1 Fault
202B RxPDO2 Fault
202C RxPDO3 Fault

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

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

* Note that the highest bit in Object 1400/1 has to be deactivated first in order to enable the correct write-access for Object 1400/2.

9.2.20 0x1600/n, 0x1601/n, 0x1602/n, RxPDO Mapping Parameters

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

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

0x1600/0 = 0 = no objects mapped

0x1600/0 = 1…8 = 1…8 objects mapped

Mapping Entry:

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

Examples:

Mapping of *0x6040/0 Control Word* (unsigned16 = 10_{hex}) to "1st mapped obj." in RxPDO1:

0x1600/1 = 60400010

Objects and their data types are listed in Chapter 9.1 "Tabular Object Overview".

Default Mapping

RxPDO1	0x1600/0	0x1600/1	0x1600/2	0x1600/3…8
	2	60 40 00 10 Controlword	60 42 00 10 target velocity	0x00000000
RxPDO2	0x1601/0			
	0	No mapping		
RxPDO3	0x1602/0			
	0	No mapping		

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

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

* Note, that the highest bit in Object 1400/1 has to be deactivated first, in order to enable the correct write-access to Object 1600/n. Also pay attention to the mapping procedure described in the following:

Mapping Procedure

The mapping process requires five steps:

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

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

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

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

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

In the example TxPDO 0x1600 was used. The same procedure can be used for 0x1601 and 0x1602. In these cases replace 0x1400 with 0x1401 and 0x1402 respectively.

Example (Node ID = 1):						
	COB ID	Control Byte	Index LSB MSB	Subindex Subindex	Data LSB …	Data … MSB
Step 1:	601	23	00 14	01	01 02	00 80
Response	581	60	00 14	01	00 00	00 00
Step 2:	601	2F	00 16	00	00 00	
Response	581	60	00 16	00	00 00	00 00
Step 3.1:	601	23	00 16	01	10 00	42 60
Response	581	60	00 16	01	00 00	00 00
Step 3.2	601	23	00 16	02	10 00	40 60
Response	581	60	00 16	02	00 00	00 00
Step 3.3	601	23	00 16	03	08 00	60 60
Response	581	60	00 16	03	00 00	00 00
Step 4:	601	2F	00 16	00	03 00	
Response	581	60	00 16	00	00 00	00 00
Step 5:	601	23	00 14	01	01 02	00 00
Response	581	60	00 14	01	00 00	00 00

Mapping Result

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

The example above shows the necessary telegrams with the corresponding responses of the device. The mapping is only stored in RAM and is therefore lost after a power restart or power failure. In order to store the mapping to EEPROM (power-fail safe), see Chapter 9.2.11.

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

9.2.21 0x1800/n, 0x1801/n, 0x1802/n, TxPDO Communication Parameters

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

TxPDO Communication Parameters:
0x1800/n TxPDO1
0x1801/n TxPDO2
0x1802/n TxPDO3

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 exists / is valid
1 = PDO does not exist / invalid

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 **every** "nth" SYNC.
~~*241 … 251*~~	Reserved	Value not allowed.
252	synchronous/RTR	Update TxPDO-data 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 und 255 are handled identically. TxPDO-data are sent, on data change or after the "Event Time".

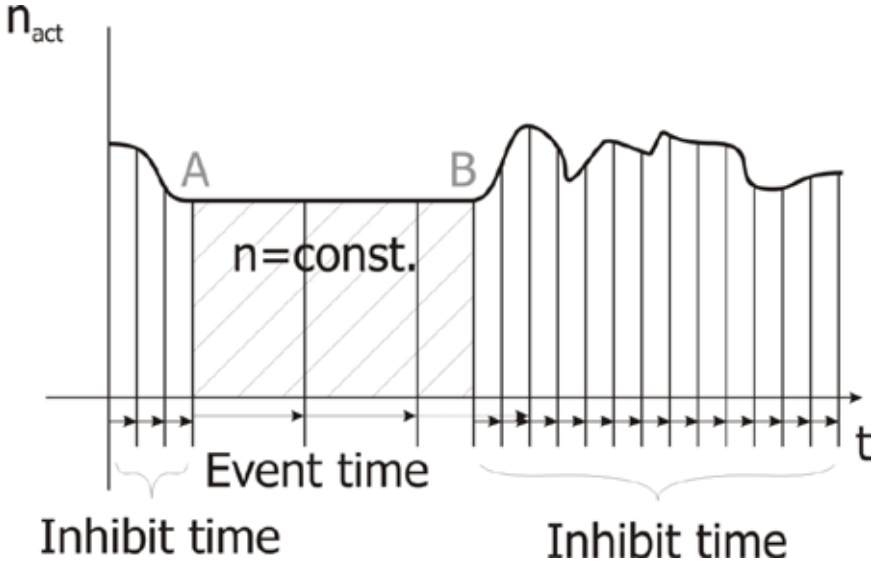
Inhibit Time:
The "Inhibit Time" is the minimum time distance between two consecutive TxPDOs for asynchronous TxPDOs. During the inhibit time, the TxPDO is not send again. Therefore a value change occurring in this time is send earliest after the inhibit time has elapsed. The value range is 0…65535.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

Note:
The internal time resolution for the "Inhibit Time" is in milliseconds. An inhibit time value = 37 is truncated to 30 [3.7 ms → 3 ms].
Values less then 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 TxPDOs data. The value range is 0…65535.

The "Event Time" is set in millisecond, e. g. a value of 2000 means 2000 ms.

Example of 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.

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

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

* Note, that the highest bit in Object 1800/1 has to be deactivated first, in order to enable the correct write-access to Object 1800/2

9.2.22 0x1A00/n, 0x1A01/n, 0x1A02/n, TxPDO Mapping Parameters

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

TxPDO-Mapping-Parameters
0x1A00/n TxPDO1
0x1A01/n TxPDO2
0x1A02/n TxPDO3

0x1A00/0 = 0 = no Object mapped

0x1A00/0 = 1 … 8 = 1 … 8 Objects mapped

Mapping Entry:

MSB			LSB
Object Index		**Sub-Index**	**Length (no. of bits)**
High byte	Low byte	si	ll

Example:

Mapping of *0x6041/0 Status Word* (unsigned16) to "1st mapped obj." in TxPDO1:

0x1A00/1 = 0x60410010

Default Mapping

TxPDO1	0x1A00/0	0x1A00/1	0x1A00/2	0x1A00/3…8
	2	*0x6041* statusword	*0x6044* control effort	0x00000000
TxPDO2	0x1A01/0			
	0	No mapping		
TxPDO3	0x1A02/0			
	0	No mapping		

Example*:

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

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

* Note, that the highest bit in Object 1800/1 has to be deactivated, in order to enable correct write-access to Object 1A00/n. Also pay attention to the mapping procedure, described below:

Mapping Procedure
The mapping procedure requires five steps:

Step 1: Set PDO to "not valid" (*0x1800*, Subindex 1, Bit 31 = 1)

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

Step 3: Set Subindex 1 … n to the new Objects (0x1A00, Subindex 1..n = new object)

Step 4: Set Subindex 0 to the number of mapped Objects (new mapping activated, 0x1A00, Subindex 0 = n)

Step 5: Set PDO "valid" (*0x1800*, Subindex 1, Bit 31 = 0)

In the example above TxPDO 0x1A00 was used. The same procedure can be used for 0x1A01 and 0x1A02. In these cases replace *0x1800* with *0x1801* and *0x1802* respectively.

Example (Node ID = 1):

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

Mapping Result			
Control effort (*0x6044*)	Status word (*0x6041*)	Digital In Actual Value (*0x3001*)	Digital Out Actual Value (*0x3002*)
00 00	00 00	00	00

Note:
The number of mappable objects depends on the object lengths. The maximum number of bytes that can be mapped is 8.

9.3 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:

Command specifier = According to the type of access and the length of the selected parameter
Index = Parameter number + 0x2000
Sub-Index = required data set (0, 1 ... 4, 5, 6 ... 9)

Note:
The mapping of numeric data is always in integer or long data type. Values which contain decimal places are extended accordingly: (e.g. value 17.35 is transmitted as 1735).
The command specifier are described in chapter 8.3 "SDO Function".
If for writing a parameter the command specifier is not known, it can be determind by reading this parameter first. The second digit in the command specifier is identical for Read and Write access.

9.3.1 Handling of Data Sets / Cyclic Writing

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

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

Caution!

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

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

Access to the Data Sets of the Frequency Inverter

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

9.3.2 SDO Examples ("expedited" transfer only)

Write Parameter:

Client ➔ Server SDO Download (expedited)

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

Server ➔ Client Download Response ➔ Write process error-free

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

Server ➔ Client Abort SDO Transfer ➔ Write process with error

	0	1	2	3	4	5	6	7
COB ID	Control Byte	Index		Sub-Index	Data			
	0x80	LSB	MSB	0xnn	Error Code			

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

Read Parameter:

Client ➔ Server SDO Upload (expedited)

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

Server ➔ Client Upload Response ➔ Read process error-free

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

Server ➔ Client Abort SDO Transfer ➔ Read process with error

	0	1	2	3	4	5	6	7
COB ID	Control Byte	Index		Sub-Index	Data			
	0x80	LSB	MSB	0xnn	Error Code			

If an error occurs during the reading process, the corresponding error code is indicated in Bytes 4 … 7. Refer to Chapter 8.3.3 "Error Code Table" for the list of error codes of the SDO Abort.

9.3.3 Examples of Writing Parameters

Write Parameter *Rated Speed* **372** (type uint) in Data Set 2 with parameter value 2980.
Index = 372 + 0x2000 = 0x2174, Value = 2980 = 0x0BA4

Client ➔ Server SDO Download (expedited)

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

Write Parameter *Warning Limit Tk* **407** (type int) in Data Set 0 with Parameter Value -15.
Index = 407 + 0x2000 = 0x2197, Value = -15 = 0xFFF1

Client ➔ Server SDO Download (expedited)

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

Writing Parameter *Fixed Frequency 1* **480** (Type long) in Data Set 1 with Parameter Value 100,00 Hz.
Index = 480 + 0x2000 = 0x21E0, Value = 10000 = 0x00002710

Client ➔ Server SDO Download (expedited)

COB ID	0	1	2	3	4	5	6	7
0x601	Control Byte	Index		Sub-Index	Data			
	0x23	0xE0	0x21	0x01	0x10	0x27	0x00	0x00

Writing Parameter *Fixed Frequency 2* **480** (Type long) in Data Set 3 with Parameter Value -50,00 Hz.
Index = 480 + 0x2000 = 0x21E0, Value = -5000 = 0xFFFFEC78

Client ➔ Server SDO Download (expedited)

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

If an error occurs during the writing process, the corresponding error code is indicated in Bytes 4 … 7. Refer to Chapter 8.3.3 "Error Code Table" for the list of error codes of the SDO Abort.

9.3.4 Examples of Reading Parameters

Read Parameter *Rated Speed* **372** (type uint) in Data Set 2 with the current parameter value 1460.
Index = 372 + 0x2000 = 0x2174, Value = 1460 = 0x05B4

Client ➔ Server SDO Upload (expedited)

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

Server ➔ Client Upload Response

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

Read Parameter *Warning Limit Tk* **407** (type int) in Data Set 0 with the current parameter value -5.
Index = 407 + 0x2000 = 0x2197, Value = -5 = 0xFFFB

Client ➔ Server SDO Upload (expedited)

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

Server ➔ Client Upload Response

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

Read Parameter *Fixed Frequency 1* **480** (type long) in Data Set 1 with the current parameter value 75,00 Hz.
Index = 480 + 0x2000 = 0x21E0, Value = 7500 = 0x00001D4C

Client ➔ Server SDO Upload (expedited)

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

Server ➔ Client Upload Response

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

Reading Parameter *Fixed Frequency 1* **480** (Type long) in Data Set 3 with the actual Parameter Value -10,00 Hz.

Index = 480 + 0x2000 = 0x21E0, Value = -1000 = 0xFFFFFC18

Client ➔ Server SDO Upload (expedited)

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

Server ➔ Client Upload Response

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

If an error occurs during the reading process, the corresponding error code is indicated in Bytes 4 … 7. Refer to Chapter 8.3.3 "Error Code Table" for the list of error codes of the SDO Abort.

9.4 Manufacturer Objects (0x3000 ... 0x5FFF)

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

9.4.1 0x3000/0 SYNC Jitter

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

DS301 does not include an object for the monitoring of SYNC messages. The frequency inverters monitor SYNC message jittering with Object 0x3000/0 *SYNC Jitter* (given in multiples of microseconds).

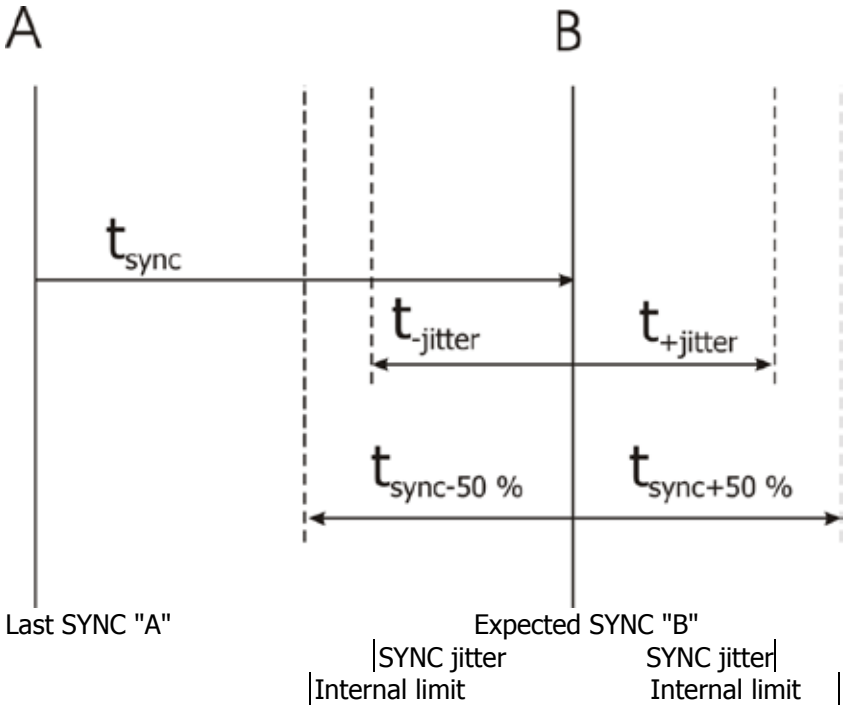
A communication error is triggered if the SYNC message does lie within the defined times:

0x1006/0 Communication Cycle Period **+/- 0x3000/0** ***SYNC Jitter***

The value of Object 0x3000/0 *SYNC Jitter* depends on time accuracy of the CANopen master. The value range is 0 ... 17000 (µs) and is limited internally to 50 % of the Communication Cycle Time (Object *0x1006/0* or the measured value).

If Object 0x3000/0 *SYNC Jitter* is set to zero, then there is no SYNC message monitoring.

If Object 0x3000/0 *SYNC Jitter* is set not equal to zero, then, monitoring of the SYNC message time is active. The jitter monitoring is independent of how the Communication Cycle Time is determined (either set with Object *0x1006/0* or by measurement).



Example:

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

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

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

9.4.2 0x3001/0 Digital In Actual Value

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

Object 0x3001 *Digital In actual value* displays, like Parameter *Digital Inputs* **250,** the current status of the Digital Inputs and Multi-functional Input 1 (if Parameter *Operation mode* **452** is set to "3 – Digital Input").

Example:

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

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

9.4.3 0x3002/0 Digital Out Actual Value

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

Object 0x3002 *Digital Out actual value* displays – like Parameter *Digital Outputs* **254** – the current status of the Digital Outputs (max. four) and Multi-functional Output 1 (if Parameter *Operation mode* **550** is set to "1 – Digital"). The number of digital outputs depends on the installed optional expansion modules.

Example:

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

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

9.4.4 0x3003/0 Digital Out set values (Sources for Digital Outputs)

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

With Object 0x3003 five digital sources are available for parameters, which demand an allocation 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
5	-	-	-
6	-	-	-

Digital outputs use these sources as operation modes 90 … 94 *Obj 0x3003 DigOut 1 … 5* and inverted as 190 … 194 *inv. Obj 0x3003 DigOut 1 … 5* (see e. g. Parameter *Operation Mode OUT1D (X13.5)* **531**). The mapping of the these object bits to the output is arbitrary.

Example:

Function	Parameter No.	Selection List (excerpt)
Operation mode OUT3D (X11.6)	533	0 - OFF (AUS) 1 - Ready or Standby Signal 2 - Run Signal 43 - External Fan 90 - Obj 0x3003 Digout 1 91 - Obj 0x3003 Digout 2 92 - Obj 0x3003 Digout 3 93 - Obj 0x3003 Digout 4 94 - Obj 0x3003 Digout 5 143 - inv. External Fan 190 - inv. Obj 0x3003 Digout 1 191 - inv. Obj 0x3003 Digout 2 192 - inv. Obj 0x3003 Digout 3 193 - inv. Obj 0x3003 Digout 4 194 - inv. Obj 0x3003 Digout 5

The sources 810…814 *Obj 0x3003 DigOut 1 … 5* can be selected directly from the parameter's choice list. This can be used, for example, for direct setting of Boolean inputs.

Example:

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

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

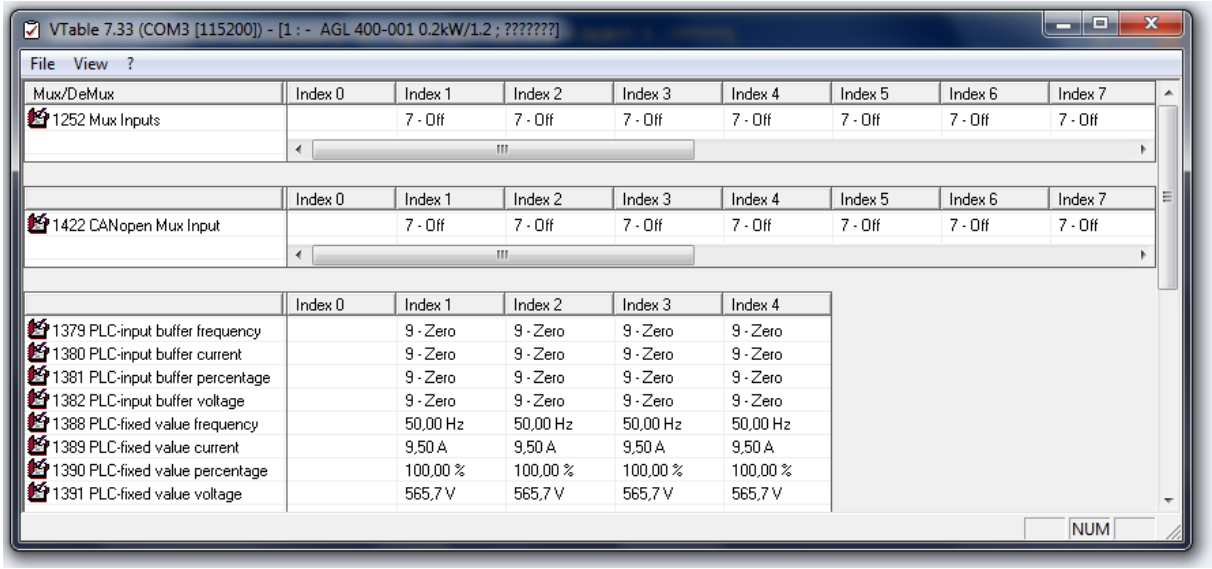
9.4.5 0x3004/0 Boolean Mux

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

With object 0x3004 up to 16 boolean values can be read in a compressed manner. Each bit in 16 bit object 0x3004 displays the actual value of the connected boolean source.

Note:
The bit numbers 0 … 15 correspond to the Index numbers 1 … 16!

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



The default value is "7 – Off".

Example:

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

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

9.4.6 0x3005/0 Boolean DeMux

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

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

Object 3005		
Bit No.	**Source No.**	**Source Name**
0	832	Obj. 0x3005 Demux Out 1
1	833	Obj. 0x3005 Demux Out 2
2	834	Obj. 0x3005 Demux Out 3
3	835	Obj. 0x3005 Demux Out 4
4	836	Obj. 0x3005 Demux Out 5
5	837	Obj. 0x3005 Demux Out 6
6	838	Obj. 0x3005 Demux Out 7
7	839	Obj. 0x3005 Demux Out 8
8	840	Obj. 0x3005 Demux Out 9
9	841	Obj. 0x3005 Demux Out 10
10	842	Obj. 0x3005 Demux Out 11
11	843	Obj. 0x3005 Demux Out 12
12	844	Obj. 0x3005 Demux Out 13
13	845	Obj. 0x3005 Demux Out 14
14	846	Obj. 0x3005 Demux Out 15
15	847	Obj. 0x3005 Demux Out 16

Example:

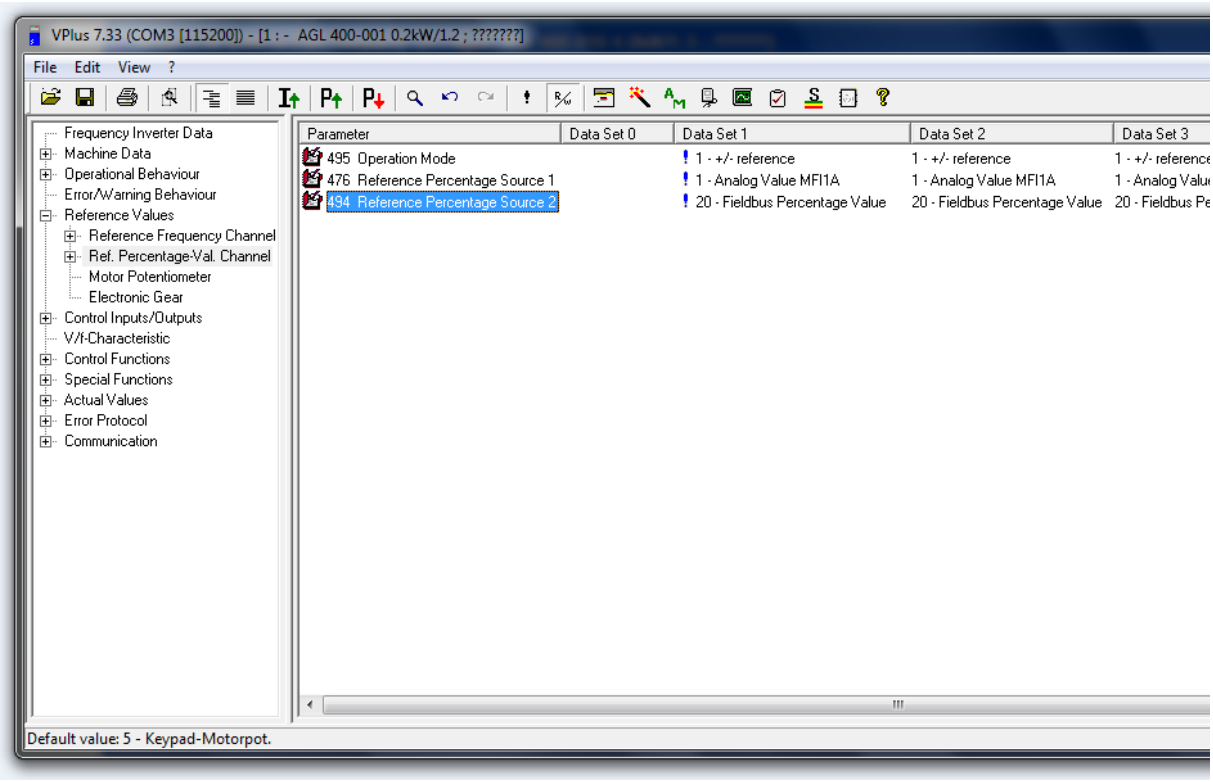
	COB ID	CB	Index	SI	Data
Read Request	601	40	05 30	00	00 00
Response	581	4B	05 30	00	05 00
Write Access	601	2B	05 30	00	20 00
Response	581	60	05 30	00	00 00

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

9.4.7 0x3006/0 Percentage set value

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

With Object 0x3006 a percentage source can be written.
The percentage reference value passed over with Object Percentage Reference Source is set with the help of selection "20 – Fieldbus Percentage Reference Value" for Parameter *Reference Percentage Source 1* **476** or *Reference Percentage Source 2* **494** in the frequency inverter.



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

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

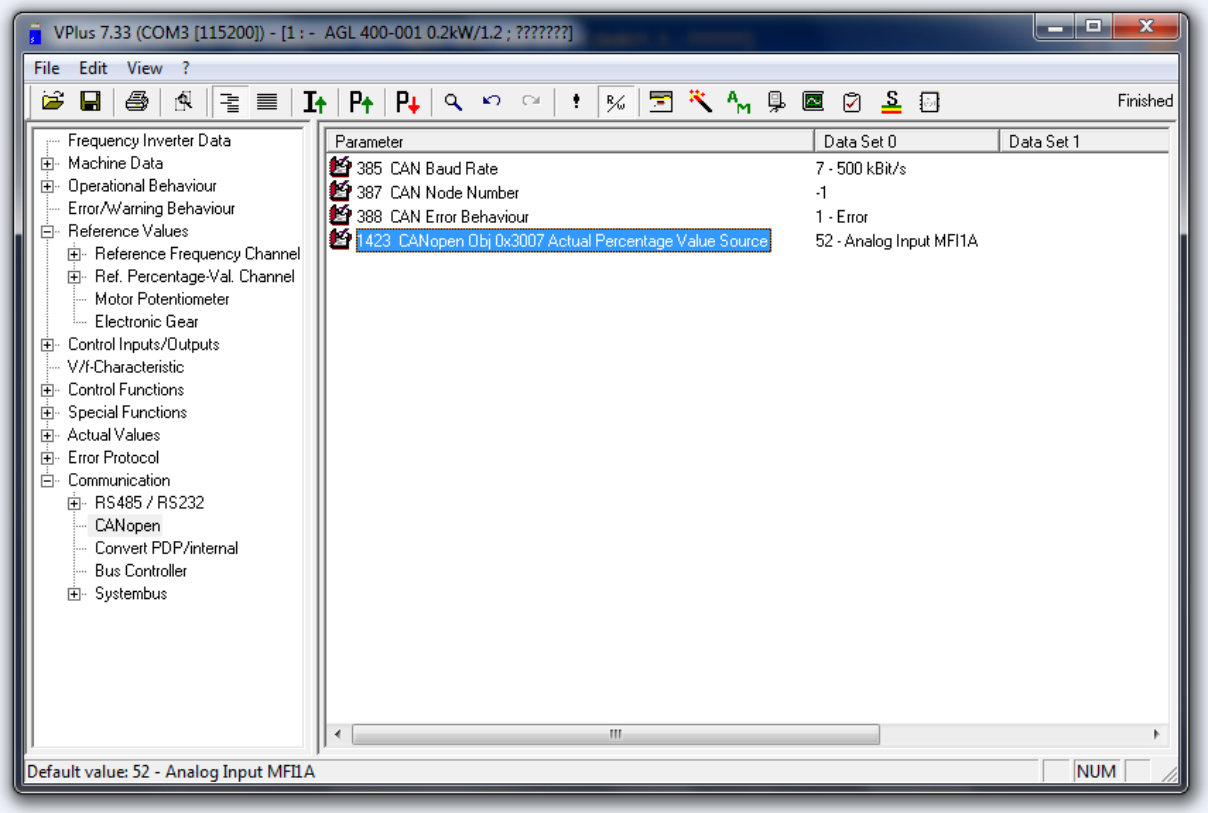
CB: Control byte SI: Sub-Index All values in hexadecimal without leading "0x"

9.4.8 x3007/0 Percentage actual value

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

Object 0x3007 displays the actual value of the percentage source, which is selectable via Parameter *CANopen Obj0x3007 Actual Percentage Value Source* **1423**.
The default value is selected: 52 – Analog Input MFI1A.

- **1423 CANopen Percentage Actual Value Source**



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

Example:

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

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

9.5 Device Profile Objects (0x6nnn)

This chapter contains the descriptions of the available "Device profile objects" defined by the CANopen standard DS402.

9.5.1 0x6007/0 Abort Connection option code

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

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

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

The object *abort connection option code* corresponds to the inverter parameter *CAN Error behavior* **388**.

CAN Error behavior **388**	**0x6007**
0	0
1	1
2	2
3	3
4	-1
5	-2

Note:
Writing Parameter *CAN Error Behavior* **388** and writing Object 0x6007 has the same effect. If object 0x6007 was written and then a save parameters command (Object *0x1010*) processed, the value of 0x6007 is stored in non volatile memory. After the next power on of the inverter the previously set value for 0x6007 is reactivated and overwrites the setting of Parameter *CAN Error behavior* **388**.

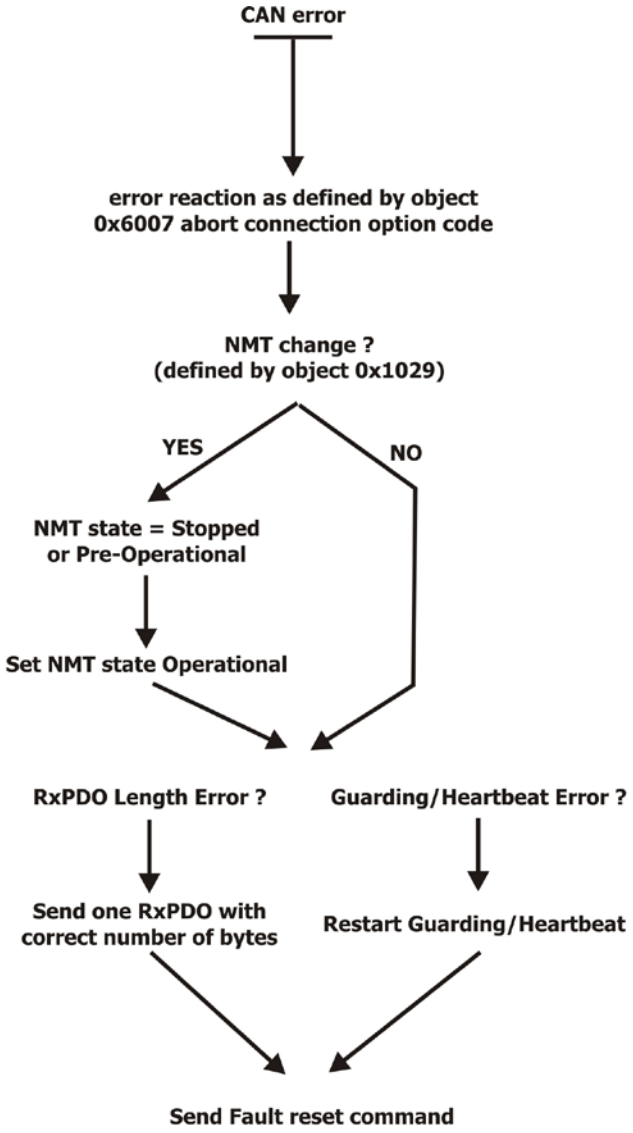
Note:
The behavior described above is ONLY relevant if parameter *Local/Remote* **412** is set to 1 "Control via Statemachine".
Otherwise the functional behavior is different. If *abort connection option code* is set to 0 "Operating point is maintained", there is no reaction. For EVERY other setting the inverter reacts with an immediate change to state "fault" if a CAN error occurs.

Attention!
The behavior for fault reset corresponds to object *0x1029/n Error Behavior*. Depending on the setting of object *Error behavior,* the NMT state may change (leaving NMT state operational). In this case the NMT state must be set to operational again before a fault reset command is accepted by the inverter.
In addition, the error reason must also be reset. E.g. for a guarding error, guarding must be started again before a fault reset command is accepted by the inverter. In the case of a RxPDO length error, a RxPDO frame with the correct number of bytes must be received before new RxPDO data is accepted.

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

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

Typical recovery sequence after a CAN-Error:



Warning! With the setting of object 0x6007 *abort connection option code* = 0 (no reaction), the inverter does not react to any CAN error and remains in its current state (e. g. drive running).
It is strongly recommended to use a setting for object 0x6007 *abort connection option code* that forces the inverter into the Fault state (setting = 1, -1 or -2).

9.5.2 0x603F/0 Error Code

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

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

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

Error Messages				
Inverter Error		**DS402 Error Code**		**Meaning**
F00	xx	00	00	No error has occurred.
Overload				
F01	xx	23	10	Frequency inverter has been overloaded
Heat Sink				
F02	xx	42	10	Heatsink temperature outside the temperature limits
Inside				
F03	xx	41	10	Inside temperature outside the temperature limits
Motor Connection				
F04	xx	43	10	Motor temperature too high or sensor defective
Output Current				
F05	xx	23	40	Motor phase current above the current limit
DC-Link Voltage				
F07	xx	32	10	DC-Link voltage outside the voltage range
Electronic Voltage				
F08	xx	51	11	Electronic voltage outside the voltage range
Motor Connection				
F13	xx	23	30	Earth fault on frequency inverter output
Generic Error				
Fyy	xx	10	00	Other error messages

If the value 1000 (= generic error) appears as the error code, then the inverter error code can be read via parameter *Current error* **260** (unsigned16). This parameter contains the error code in the product-internal format.
The assignment table of the error code to the corresponding messages can be taken from the operating instructions.

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

Example:

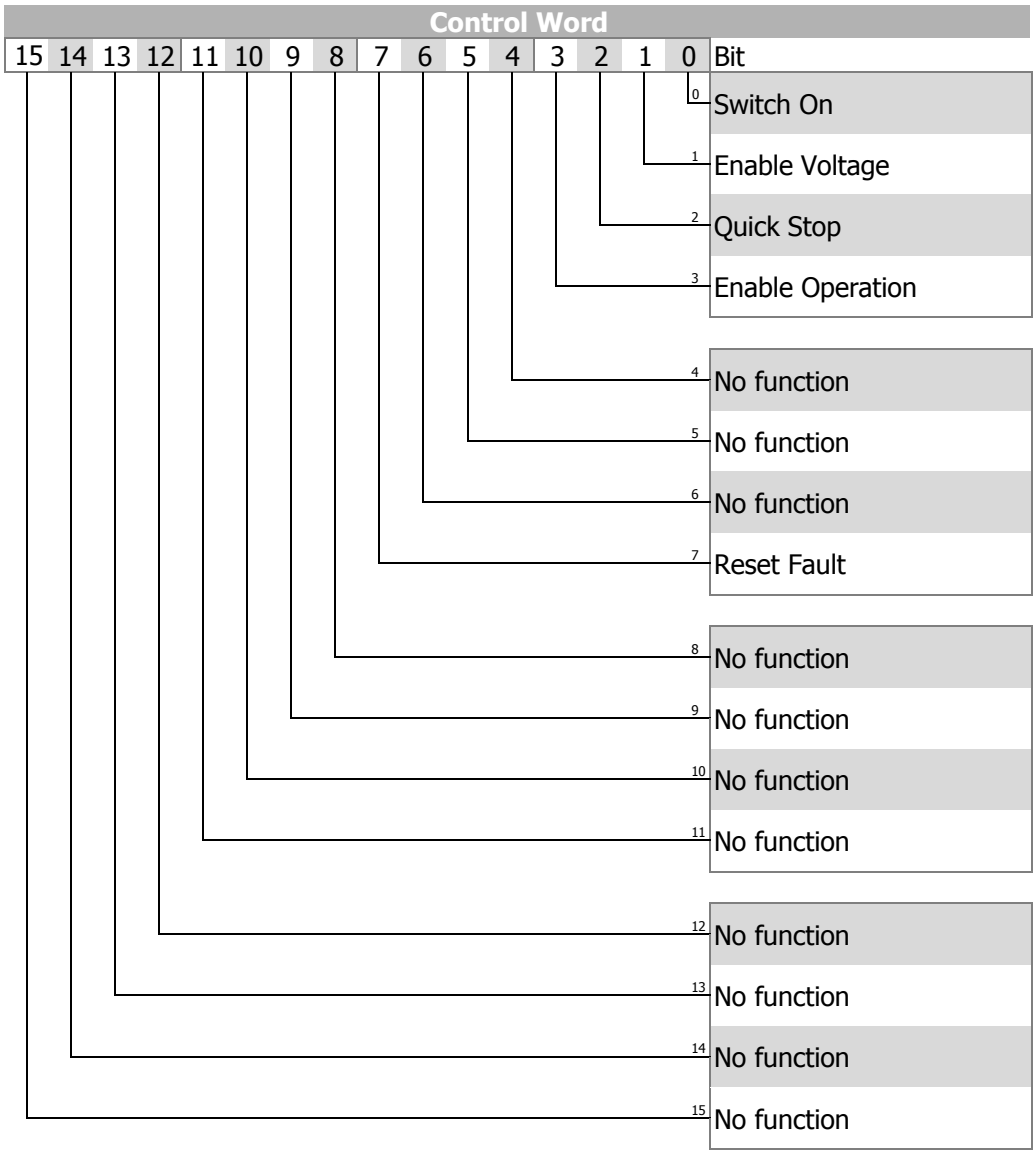
	COB ID	CB	Index	SI	Data
Read Request	601	40	3F 60	00	00 00
Response	581	4B	3F 60	00	00 00

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

9.5.3 0x6040/0 Control Word

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6040	0	Controlword	Unsigned16	rw	Rx	0

Object 0x6040/0 *Control Word* is relevant to the inverter remote state machine whenever Parameter *LocalRemote* **412** is set to 1 (remote state machine).
The Object 0x6040/0 *Control Word* is linked to the internal Parameter *Controlword* **410**.
Please check also chapters 10.2 "Control via Statemachine" and 10.1 "Control via Contacts/Remote-Contacts". When operating with CANopen use 0x6040/0 *Control Word* instead of Parameter *Control Word* **410**.



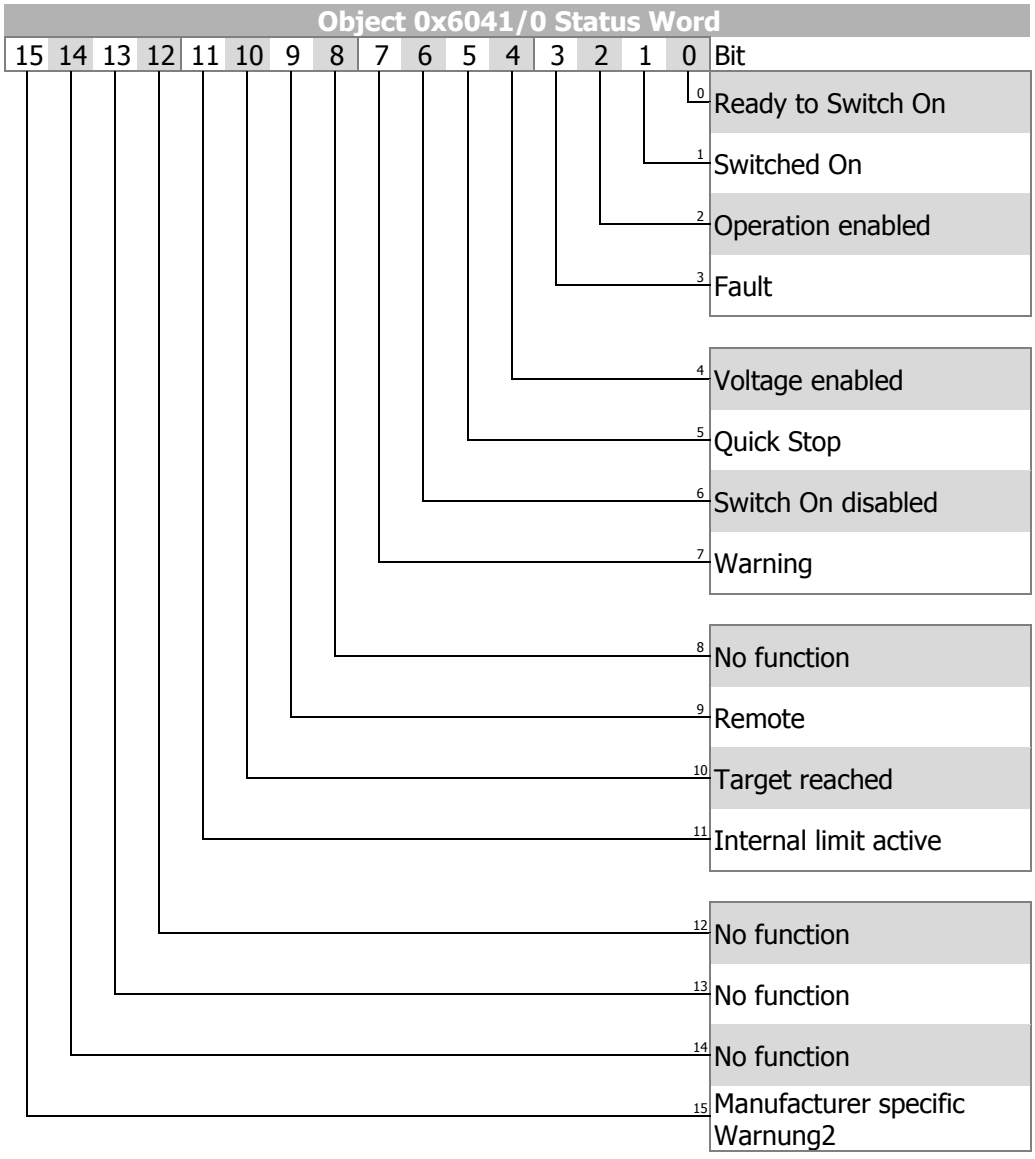
Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	40 60	00	00 00
Response	581	4B	40 60	00	01 00
Write Access	601	2B	40 60	00	06 00
Response	581	60	40 60	00	00 00

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

9.5.4 0x6041/0 Status Word

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6041	0	Statusword	Unsigned16	ro	Tx	

Object 0x6041/0 *Status Word* displays the current state of frequency inverter.
The Object 0x6041/0 *Status Word* is linked to the internal Parameter *Status Word* **411.**
Please check also chapters 10.2 "Control via Statemachine" and 10.1 "Control via Contacts/Remote-Contacts". When operating with CANopen use 0x6040/0 *Status Word* instead of Parameter *Status Word* **411**.



Example:

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

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

9.5.5 0x6042/0 Target Velocity

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

Object *target velocity* is the speed reference value for the frequency inverter in velocity mode. *Target velocity* is interpreted as a speed with the unit RPM. The inverter's internal reference frequency is calculated from the target velocity in RPM taking into account parameter *No. of Pole Pairs* **373** (value of P.373 data set 1).

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

The reference value *Target-velocity* is product-internally connected to the Fieldbus reference value..
The speed reference value provided by Object 0x6042 is set in the frequency inverter with the help of selection "20 – Fieldbus Reference Value" for Parameter *Reference Frequency Source 1* **475** or *Reference Frequency Source 2* **492**.

Note:
Parameter *No. of Pole Pairs* **373** has four different data sets.
Some applications often have more than one motor connected to the inverter (only one at a time, switched over by contactor). These motors may have a different no. of pole pairs. The entry in *No. of Pole Pairs* **373** is then different in the four data sets. After change-over of the motor, the object *target velocity* must be written at least once in order to recalculate the internal reference frequency of the inverter using the correct no. of pole pairs.

Example:

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

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

9.5.6 0x6043/0 Target velocity demand (Ramp Output)

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

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

Example:

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

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

9.5.7 0x6044/0 Control effort (current Speed)

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

Object *control effort* is the actual speed of the drive in RPM. The object has the same notation as the object *target velocity* and can be read as an actual value. For calculating *control effort* the parameter *No. of Pole Pairs* **373** (value of P.373 in the active data set) is taken into account in the same way as described for object *target velocity*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	44 60	00	00 00
Response	581	4B	44 60	00	DE 01

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

9.5.8 0x6046/n Velocity min max amount

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6046	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Velocity min amount (RPM)	Unsigned32	rw	No	See text
	2	Velocity max amount (RPM)	Unsigned32	rw	No	See text

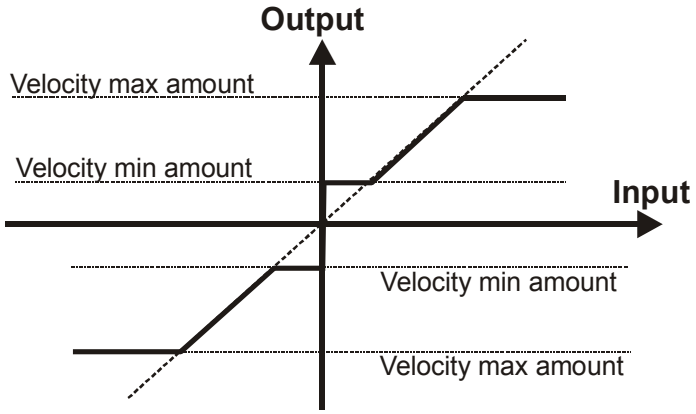
Object *velocity min max amount* comprises the Sub-Index 01 = *velocity min amount* und Sub-Index 02 = *velocity max amount*.

The unit of 0x6046/1 *velocity min amount* is in RPM (positive values only). Writing to object 0x6046/1 *velocity min amount* automatically generates a write command to parameter *Minimum Frequency* **418** (data set 5, all data sets in RAM only !). The value of 0x6046/1 *velocity min amount* is converted internally to a frequency value, taking into account the parameter *No. of Pole Pairs* **373** (in data set 1).

The unit of 0x6046 *velocity max amount* is in RPM (positive values only). Writing to object 0x6046/2 *velocity max amount* automatically generates a write command to parameter *Maximum Frequency* **419** (data set 5, all data sets in RAM only !). The value of 0x6046/2 *velocity max amount* is converted internally to a frequency value, taking into account parameter *No. of Pole Pairs* **373** (in data set 1).

If the Reference Value of Object *0x6042/0 Target Velocity* is less than the Object Value 0x6046/1 *velocity min amount* or greater than 0x6046/2 *velocity max amount*, then *0x6042/0 Target Velocity* is limited accordingly.

No.	Object	Min.	Max.
0x6046/1	Velocity min amount (min^{-1})	0	32767
0x6046/2	Velocity max amount (min^{-1})	0	32767



Note:
If Objects 0x6046/1 or 0x6046/2 were written and then a save parameters command (Object *0x1010*)processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of parameters *Minimum Frequency* **418** und *Maximum Frequency* **419**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	46 60	01	00 00 00 00
Response	581	43	46 60	01	00 00 00 00
Write Access	601	23	46 60	01	DC 05 00 00
Response	581	60	46 60	01	00 00 00 00

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

9.5.9 0x6048/n Velocity acceleration

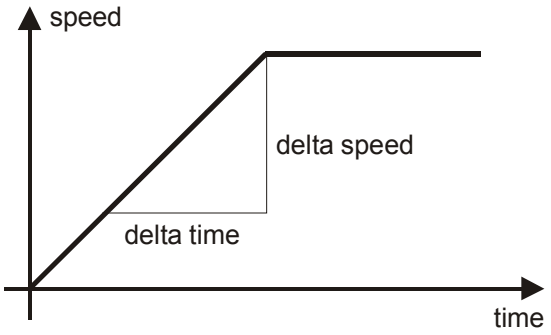
Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6048	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

The acceleration in velocity mode is set with object *velocity acceleration*. The object *velocity acceleration* consists of *delta speed* in RPM and *delta time* in seconds.

The gradient of the frequency in the acceleration period is written to parameters *Acceleration (clockwise)* **420** and *Acceleration (anti-clockwise)* **422** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.420 and p.422 are converted internally to a frequency/sec value, taking into account parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects delta time or delta speed.

No.	Object	Min.	Max.
0x6048/1	Delta speed (RPM)	1	32767
0x6048/2	Delta time (sec)	1	65535



Note:
If objects 0x6048/1 or 0x6048/2 were written and then a save parameters command (Object *0x1010*) processed, the object values are stored in volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of Parameters *Acceleration Clockwise* **420** and *Acceleration Anticlockwise* **422**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	48 60	01	00 00 00 00
Response	581	43	48 60	01	96 00 00 00
Write Access	601	23	48 60	01	50 50 00 00
Response	581	60	48 60	01	00 00 00 00

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

9.5.10 0x6049/n Velocity deceleration

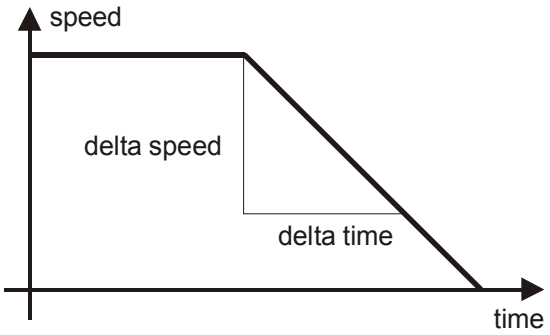
Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6049	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

The deceleration in velocity mode is set with object *velocity deceleration*. The object *velocity deceleration* consists of *delta speed* in rpm and *delta time* in seconds.

The gradient of the frequency in the deceleration period is written to parameters *Deceleration (clockwise)* **421** and *Deceleration (anti-clockwise)* **423** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.421 and p.423 are converted internally to a frequency/sec value, taking into account the parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects delta time or delta speed.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x6049/1	Delta speed (RPM)	1	32767
0x6049/2	Delta time (sec)	1	65535



Note:
If objects 0x6049/1 or 0x6049/2 were written and then a save parameters command (Object *0x1010*) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of Parameters *Deceleration Clockwise* **421** and *Deceleration Anticlockwise* **423**.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	49 60	01	00 00 00 00
Response	581	43	49 60	01	96 00 00 00
Write Access	601	23	49 60	01	40 50 00 00
Response	581	60	49 60	01	00 00 00 00

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

9.5.11 0x604A/n Velocity quick stop

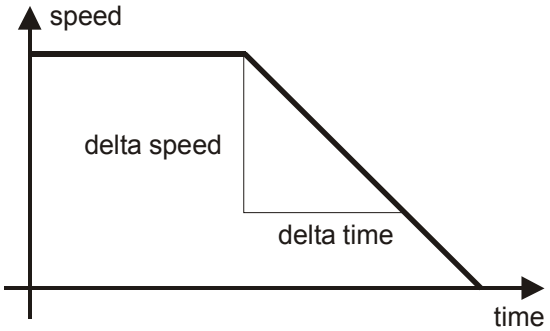
Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x604A	0	Highest sub-index supported	Unsigned8	ro	No	2
	1	Delta speed (min^{-1})	Unsigned32	rw	No	0x96
	2	Delta time (sec)	Unsigned16	rw	No	1

The quick stop deceleration in velocity mode is set with object *velocity quick stop*. Object *velocity quick stop* consists of *delta speed* in RPM and *delta time* in seconds.

The gradient of the frequency in the deceleration period is written to parameters *Emergency Stop (clockwise)* **424** and *Emergency Stop (anti-clockwise)* **425** (data set 5, all data sets in RAM only !). Both parameters are set to the same value.
The values of p.424 and p.425 are converted internally to a frequency/sec value, taking into account the parameter *No. of Pole Pairs* **373** (in data set 1 !).

The gradient is changed internally by altering the objects *delta time* or *delta speed*.

Parameter		Setting	
No.	**Object**	**Min.**	**Max.**
0x604A/1	Delta speed (min^{-1})	1	32767
0x604A/2	Delta time (sec)	1	65535



Note:
If objects 0x604A/1 or 0x604A/2 were written and then a save parameters command (Object *0x1010*) processed, the object values are stored in non volatile memory. After the next power on of the inverter, the previously set values are reactivated and overwrite the settings of Parameters *Emergency Stop (clockwise)* **424** and *Emergency Stop (anti-clockwise)* **425**.

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	4A 60	01	00 00 00 00
Response	581	43	4A 60	01	96 00 00 00
Write Access	601	23	4A 60	01	20 50 00 00
Response	581	60	4A 60	01	00 00 00 00

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

9.5.12 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 *modes of operation* is set fixed to "2 - velocity mode".

Modes of operation
2 – velocity mode

Refer to Chapter 10 "Frequency Inverter Control" for further information.

Example:					
	COB ID	CB	Index	SI	Data
Write Access	601	2F	60 60	00	02
Response	581	60	60 60	00	00

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

9.5.13 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	2

Object *Modes of operation display* acknowledges the fixed value set with *0x6060/0 Modes of operation* by displaying Operation Mode as the set fixed value of 2.

Note:
After setting *modes of operation,* the PLC must wait for this acknowledgement before sending any other command to the inverter.

For further information refer to Chapter 10 "Frequency Inverter Control".

Example:					
	COB ID	CB	Index	SI	Data
Read Request	601	40	61 60	00	00
Response	581	4F	61 60	00	02

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

9.5.14 0x6071/0 Target Torque

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

The value transmitted via Object 0x6071 is selectable as source Q.808 for various parameters (e.g. *FT Input buffer percentage* **1381**).
It is also available as Operation Mode 95 (e.g. for Parameter *Reference Percentage Source* **476**) in configurations with torque control.

A value 0x3E8 (=1000) corresponds to rated motor torque (100,0%).

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

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	71 60	00	00 00
Response	581	4B	71 60	00	00 00
Write Access	601	2B	71 60	00	64 00
Response	581	60	71 60	00	00 00

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

9.5.15 0x6077/0 Torque actual value

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6077	0	Torque actual value	Integer16	ro	Tx	

Object 0x6077 *Torque actual value* displays the torque actual value.

A value of 0x3E8 (=1000) corresponds to rated motor torque (100,0%). Please refer as well to Object *0x6071*.

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	77 60	00	00 00
Response	581	4B	77 60	00	00 00

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

9.5.16 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* displays the current actual value.

A value of 0x3E8 (=1000) corresponds to the rated motor current (100,0%).

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	78 60	00	00 00
Response	581	4B	78 60	00	00 00

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

9.5.17 0x6079/0 DC link circuit voltage

Index	Sub-Index	Meaning	Data Type	Access	Map	Def.-Val
0x6079	0	DC link circuit voltage	Unsigned32	ro	Tx	

Object 0x6079 *DC link circuit voltage* displays the voltage actual value of the in mV (see Parameter *DC-Link Voltage* **222**).

A value 0x0001 86A0 (=100 000) corresponds to 100.000 V (three decimal points).

Example:

	COB ID	CB	Index	SI	Data
Read Request	601	40	79 60	00	00 00 00 00
Response	581	43	79 60	00	CA E8 04 00

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

10 Frequency Inverter Control

- **412 Local/Remote**

In principle, the frequency inverter can be controlled with three operation modes. The operation modes can be selected with the data set changeable Parameter *Local/Remote* **412**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Factory Setting**
412	Local/Remote	0	44	44

Only operation modes 0, 1 and 2 are relevant for operation with CANopen. The remaining settings are relevant to control via the operator panel.

Operation Mode	Function
0 - Control via Contacts (Chapter 10.1)	The Start and Stop commands as well as the direction of rotation are set via digital signals.
1 - Control via Statemachine (Chapter 10.2)	The frequency inverter is controlled via the Control Word.
2 - Control via Remote-Contacts (Chapter 10.1)	The Start and Stop commands as well as the direction of rotation are set via virtual digital signals of the Control Word.

Note:
Parameter *Local/Remote* **412** is data set changeable, i.e. switching between the various operation modes can be carried out via data set selection.

- **414 Data Set Selection**

The data set selection can be carried out locally via the control contacts at the digital inputs of the frequency inverter or via the bus. Parameter *Data Set selection* **414** is used for switching the data set via the bus.

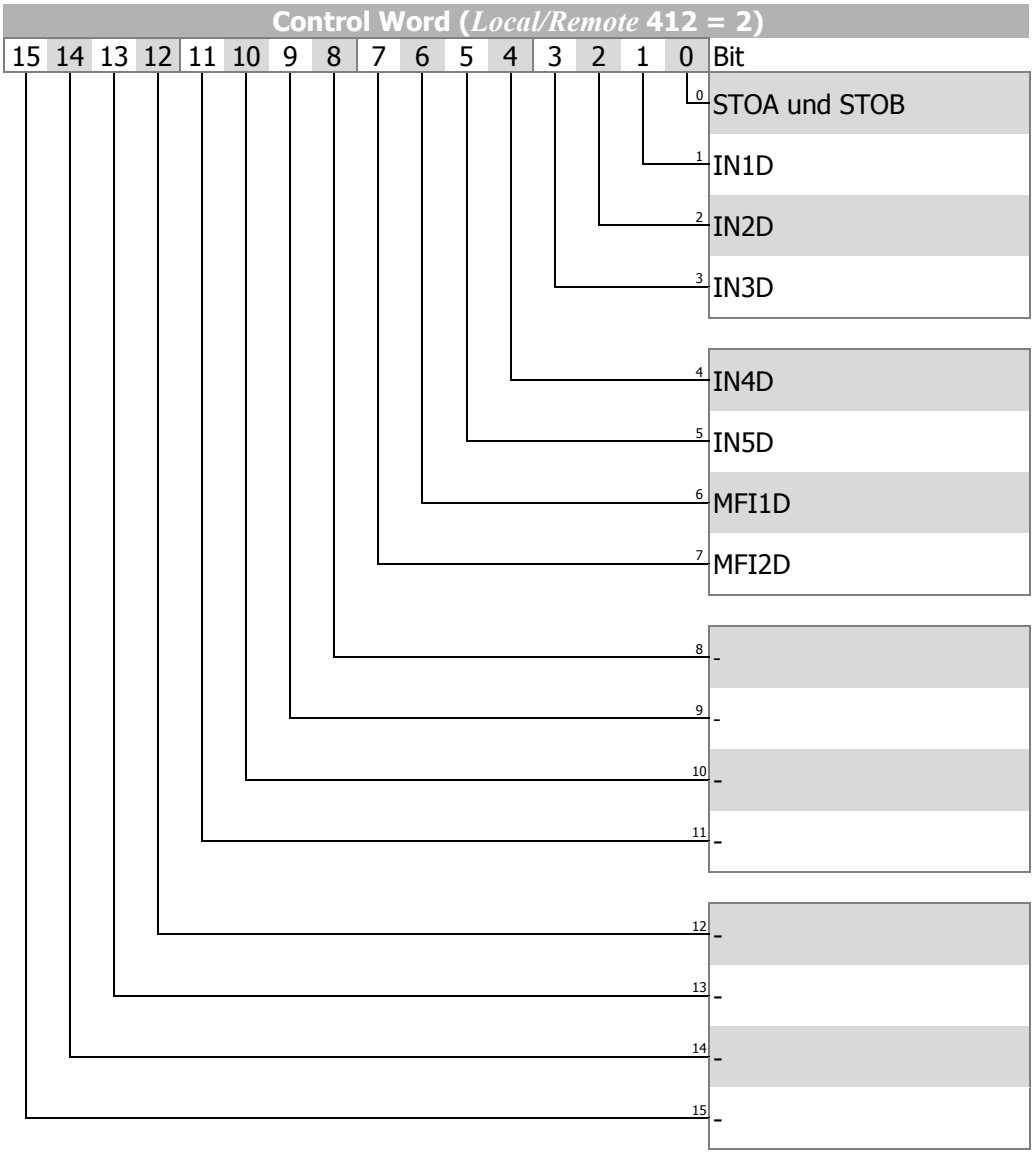
Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Factory Setting**
414	Data Set selection	0	5	0

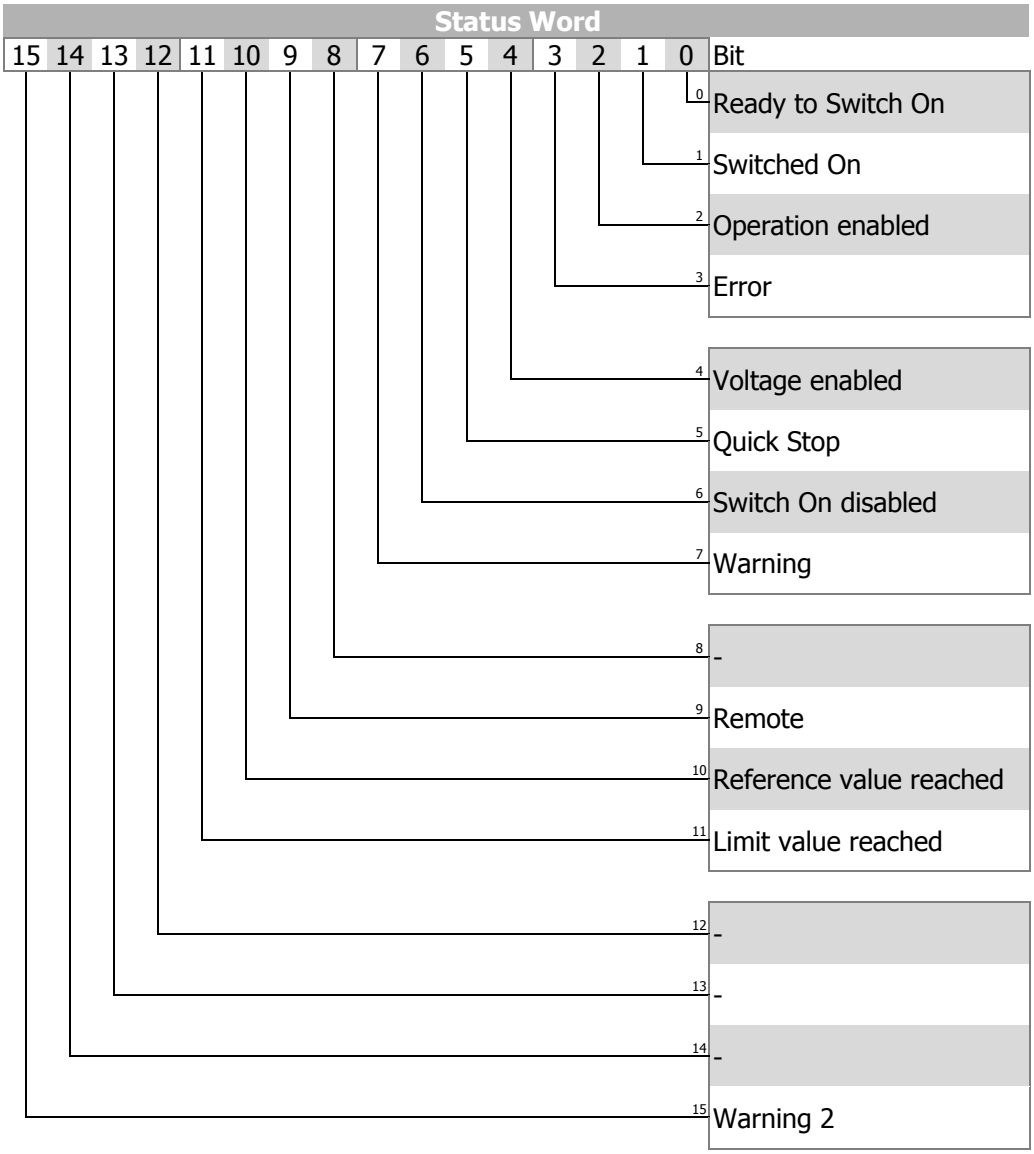
If *Data Set Selection* **414** = 0, data set switching via contact inputs is active.
If *Data Set Selection* **414** = 1, 2, 3 or 4, the corresponding data set is activated and data set switching via contact inputs is deactivated.
If *Data Set Selection* **414** = 5, then data set switching is only carried out whenever the frequency inverter is not released.

The currently selected data set can be read out with Parameter *Active Data Set* **249**. *Active data set* **249** states the activated data set with the value 1, 2, 3 or 4. This is independent of whether the data set change-over was carried out via control inputs or via *Data Set Selection* **414**.

10.1 Control via Contacts/Remote-Contacts

In the operation mode "Control via Contacts" or "Control via Remote-Contacts" (Parameter *Local/Remote* **412** = 0 or 2), the frequency inverter is controlled directly via the digital inputs or via the individual bits of the virtual signals in the *Control Word*. The meaning of these inputs is described in the operating instructions of the frequency inverter.





Note:
If operation mode "Control via Remote-Contacts" is used, then the Controller Release on STOA (Terminal X11.3) and STOB (Terminal X13.3) must be switched on **and** bit 0 of the Control Word must be set in order to start the drive.
The operation modes "Control via Contacts" and "Control via Remote-Contacts" only support the *mode of operation* "velocity mode".

Note:
The frequency inverters support an external 24 V voltage supply for the control electronics of the frequency inverter. Communication between the controlling device (PLC) and the frequency inverter is still possible even when the mains supply has been switched off.

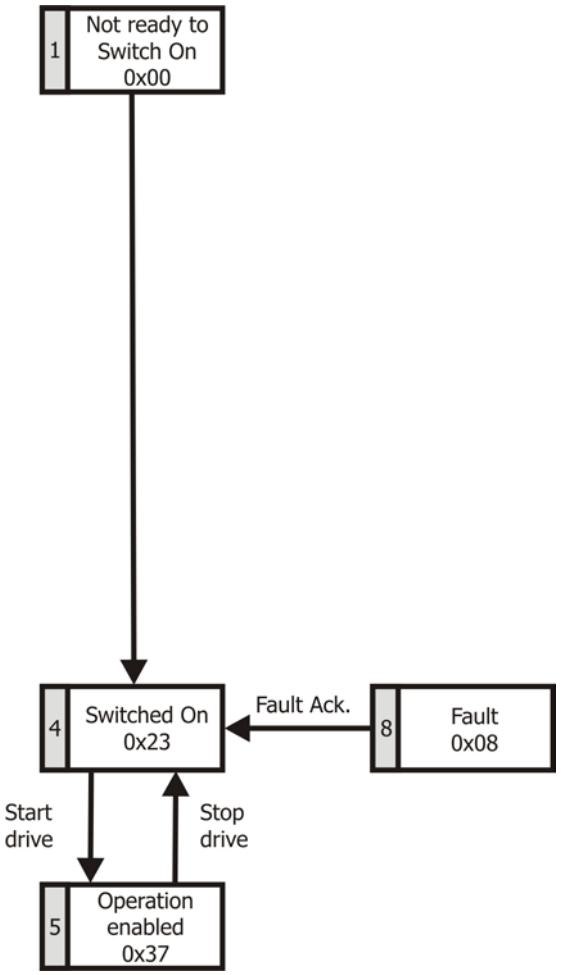
Bit 4 "Voltage enabled" in the Status Word indicates the current status of the mains supply.

Bit 4 "Voltage enabled" = 0 signals "no mains supply" and that starting the drive is not possible.

Bit 4 "Voltage enabled" = 1 signals "mains supply switched on" and drive ready for starting.

10.1.1 Device state machine

Statemachine:



Status Word	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0
Switched On	1	0	0	1	1
Operation enabled	1	0	1	1	1
Error	x	1	x	x	x

Note:
"x" means any value.

Bit 7, **Warning**, can be set at any time. It indicates a device-internal warning message. The evaluation of the present warning is carried out by reading out the warning status with Parameter *Warnings* **270**.

Bit 10 "**Reference value reached**" is set when the specified reference value is reached. In the special case of power failure regulation, the bit is also set when the power failure regulation reaches the frequency 0 Hz. For "Reference value reached" there is a hysteresis (tolerance range) which can be set via parameter *Reference Value Reached: Hysteresis* **549**.

Bit 11 "**Limit value reached"** indicates that an internal limit is active. This may be the current limit, the torque limit or the overvoltage control. All functions result in the reference value being left or not reached.

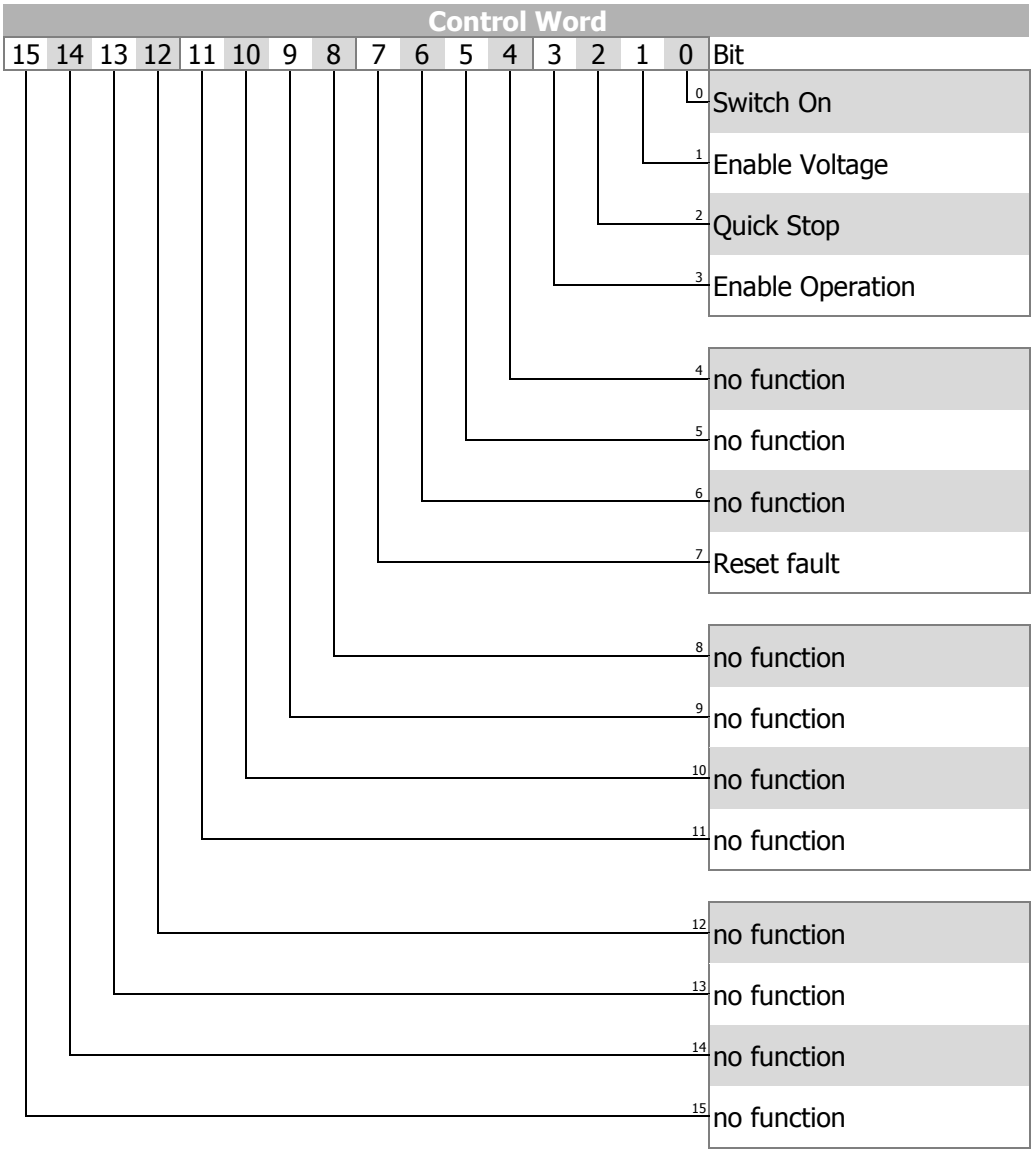
Bit 15 "**Warning 2**" signals a critical operating state which will result in a fault switch-off of the frequency inverter within a short time. This bit is set if there is a delayed warning relating to the motor temperature, heat sink/inside temperature, Ixt monitoring or mains phase failure.

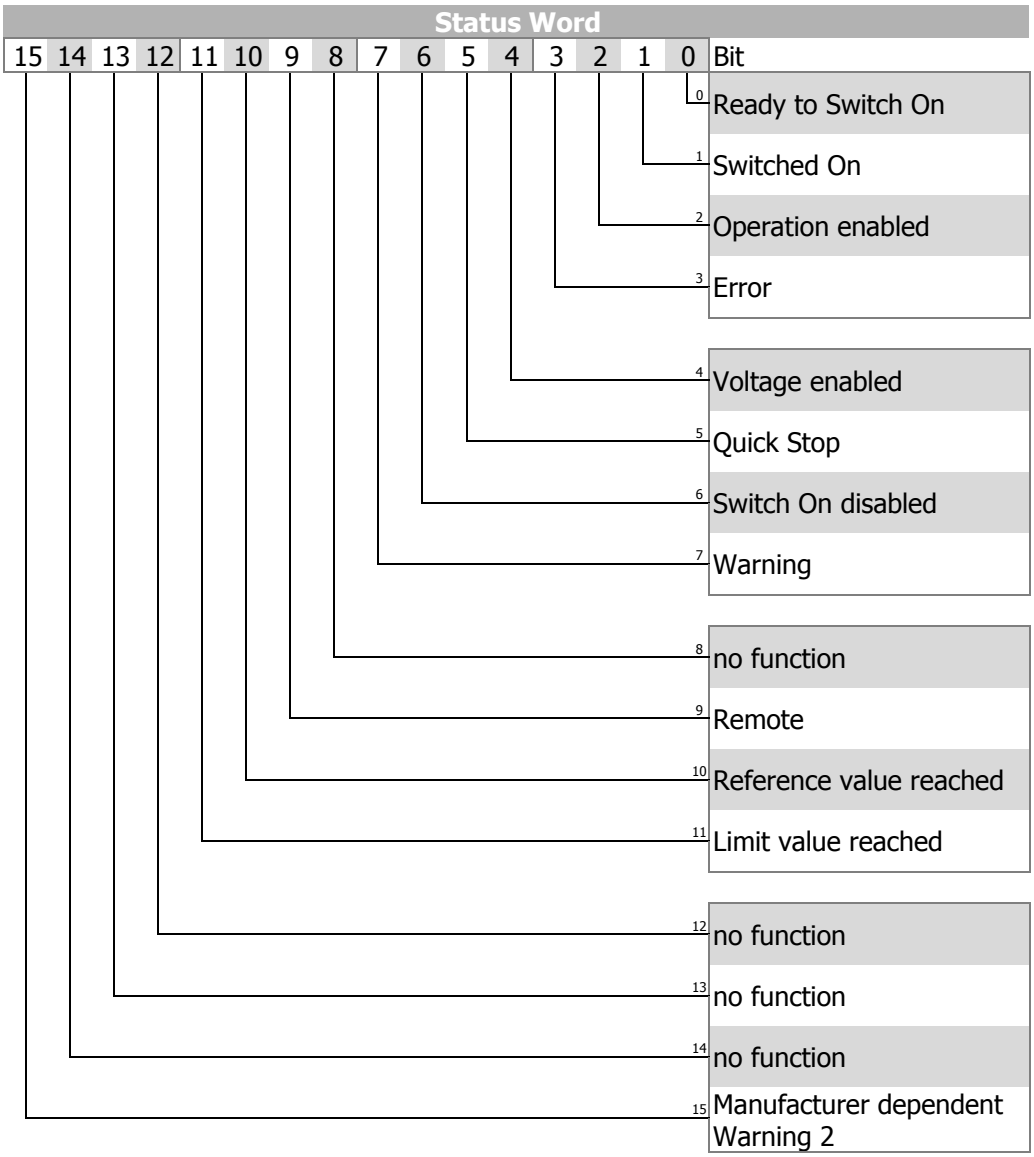
10.2 Control via Statemachine

In the operation mode "Control via Statemachine" (*Local/Remote* **412** = 1) the frequency inverter is controlled via the Control Word of the Statemachine.

Transition 4 to state "Operation enabled" is only possible if the Controller Release via STOA and STOB and one of the digital inputs for Start Right or Start Left is set.

Object *0x6040/0 Control Word* is used for the frequency inverter if Parameter *Local/Remote* **412** is set to "1 – Control via Statemachine".





Note:
Agile frequency inverters support an external 24 V voltage supply for the control electronics of the frequency inverter. Communication between the controlling device (PLC) and the frequency inverter is still possible even when the mains supply has been switched off.

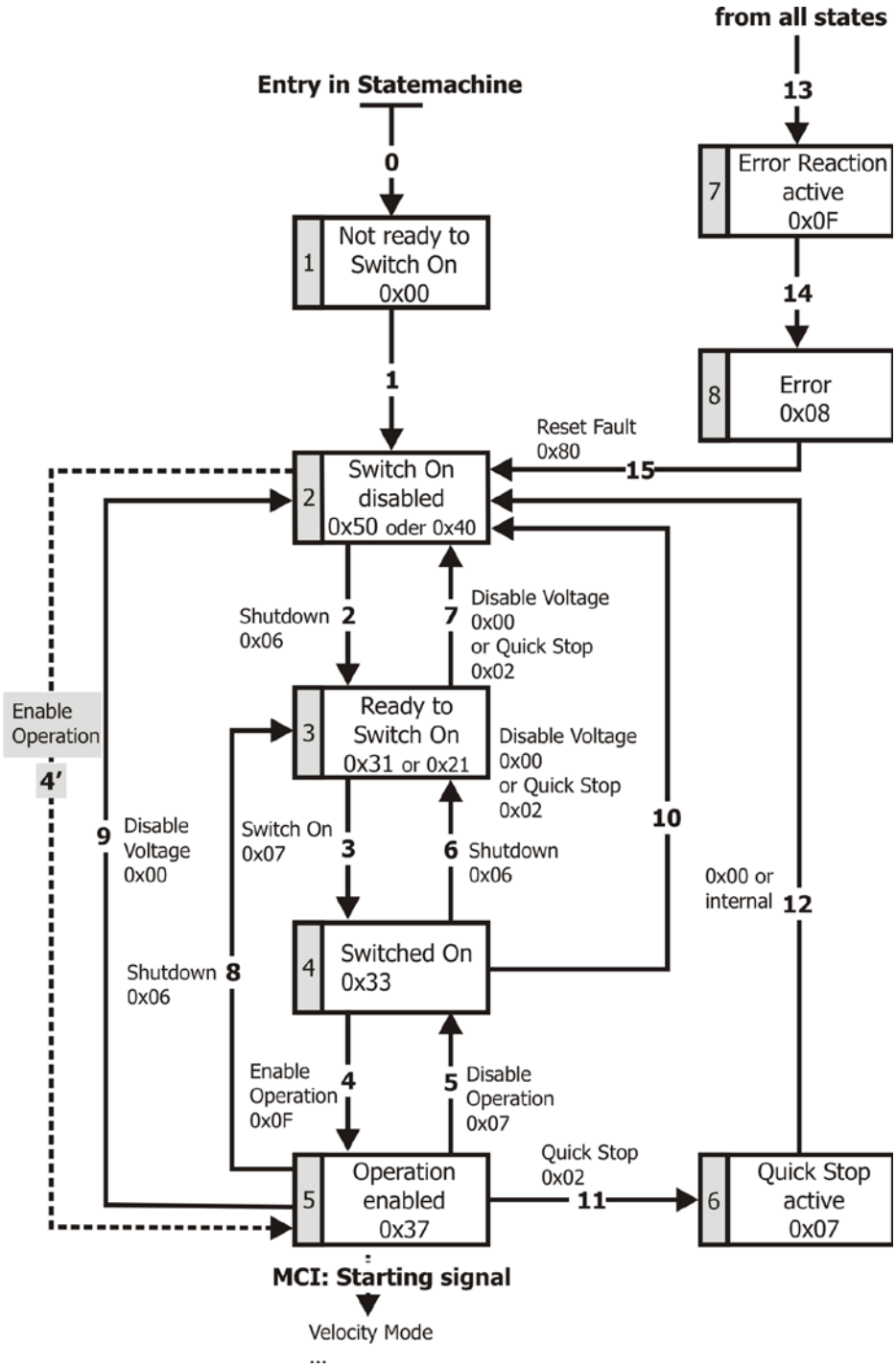
Bit 4 "Voltage enabled" in the Status Word indicates the current status of the mains supply.

Bit 4 "Voltage enabled " = 0 signals "no mains supply" and that starting the drive is not possible.

Bit 4 "Voltage enabled " = 1 signals "mains supply switched on" and drive ready for starting.

10.2.1 Statemachine diagram

Statemachine:



Control Word:

The device control commands are triggered by the following bit combinations in the Control Word.

Control Word						
Command	Bit 7 **Reset Fault**	Bit 3 **Enable Operation**	Bit 2 **Quick Stop**	Bit 1 **Enable Voltage**	Bit 0 **Switch On**	**Transitions**
Shutdown	X	X	**1**	**1**	**0**	2, 6, 8
Switch On	X	**0**	**1**	**1**	**1**	3
Switch On	X	**1**	**1**	**1**	**1**	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12

Quick Stop	X	X	**0**	**1**	X	7, 10, 11
Disable Operation	X	**0**	**1**	**1**	**1**	5
Enable Operation	X	**1**	**1**	**1**	**1**	4
Reset Fault	**0 ⇨ 1**	x	x	x	x	15

"X" means any value.

Note:
Transition 3 (Command "Switch On") is only processed if Bit 4 "Voltage enabled" in the Status Word is set.
Note:

- State transition **4'** is available and is only processed if bit no. 4 "Voltage enabled" of the Status Word is set.
- The inverter can only be controlled via the state machine if the logic linking is true. The logic inputs for Start clockwise / Start anticlockwise can be connected directly to "ON" or "OFF" (Parameters *Start-clockwise* **68** and *Start-anticlockwise* **69**).
  Digital inputs (STOA und STOB) must be set.
  Therefore this results in:
  Release: (= STOA und STOB) A**ND** (Start clockwise **OR** Start anticlockwise)

Status Word:
The Status Word reflects the operation state.

Status Word

State	Bit 6 **Switch On disabled**	Bit 5 **Quick Stop**	Bit 3 **Fault**	Bit 2 **Operation enabled**	Bit 1 **Switched On**	Bit 0 **Ready to Switch On**
Switch On disabled	**1**	X	**0**	**0**	**0**	**0**
Ready to Switch On	**0**	**1**	**0**	**0**	**0**	**1**
Switched On	**0**	**1**	**0**	**0**	**1**	**1**
Operation enabled	**0**	**1**	**0**	**1**	**1**	**1**
Quick stop active	**0**	**0**	**0**	**1**	**1**	**1**
Fault reaction active	**0**	X	**1**	**1**	**1**	**1**
Fault	**0**	X	**1**	**0**	**0**	**0**

"X" means any value.

Bit 7 **"Warning"**, can be set at any time. It indicates a device-internal warning message.
The evaluation of the warning reason is carried out by reading out the warning status with the parameter *Warnings* **270**.

Bit 9 **"Remote"**, is set if the operation mode "control via state machine" (*Local/Remote* **412** = 1) has been set **and** the hardware release is available.

Bit 10 "**Target reached"**, is set whenever the specified reference value has been reached. target reached is related to the Reference Speed Object) *0x6042/0 Target Velocity*. In the special case of power failure regulation, the bit is also set if the power failure regulation has reached the frequency 0 Hz (see operating instructions).
For "Target reached" there is a hysteresis (tolerance range), which can be set via parameter *Reference Value Reached: Hysteresis* **549** (see operating instructions).

Bit 11 **"Internal limit active"**, indicates that an internal limit is active. This can, for example, be the present current limit, the torque limit or the over-voltage limit. All of these limit functions lead to the reference value being quit or not reached.

Bit 15 "**Warning 2"**, indicates a warning which leads to a fault switch-off of the frequency inverter within a short period of time. This bit is set if there is a warning for motor temperature, heat sink/inside temperature, Ixt monitoring or mains phase failure.

10.3 Operation Mode Velocity Mode

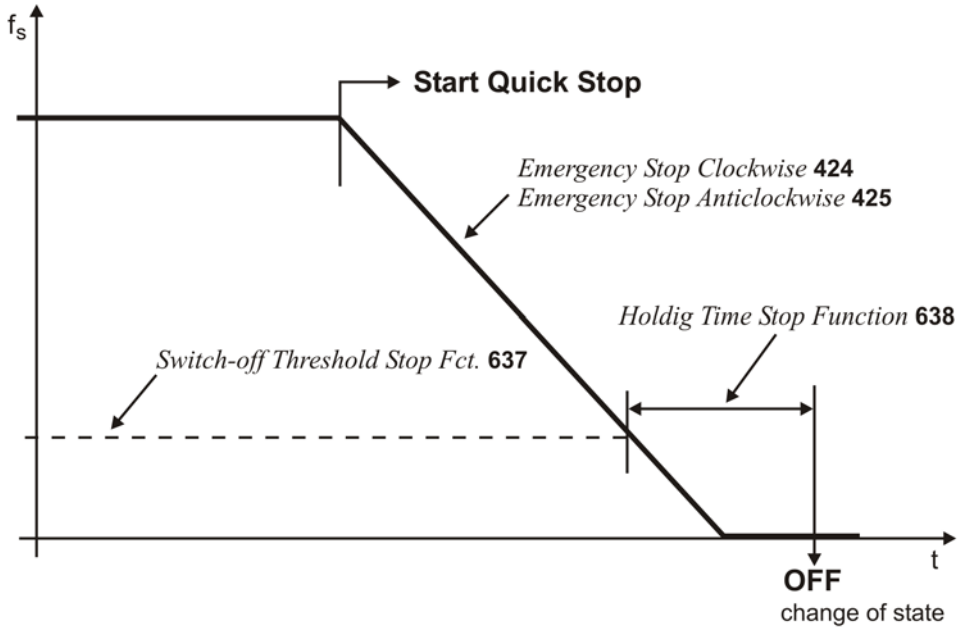
Object *0x6060/0 Modes of operation* is fixed to "2 – *velocity mode*". Object *0x6061/0 Modes of operation display* is always "2 – *velocity mode"*. These settings cannot be changed.

Related Objects:

0x6040	Controlword
0x6041	Statusword
0x6042	Target velocity
0x6043	Velocity demand
0x6044	Control effort
0x6046	Velocity min max amount
0x6048	Velocity acceleration
0x6049	Velocity deceleration
0x604A	Velocity quick stop

10.3.1 Behavior in Quick Stop

Here, the parameters *Switch-Off Threshold Stop Function* **637** (percentage of parameter *Maximum Frequency* **419**) and *Holding Time Stop Function* **638** (holding time after the value drops below the switch-off limit) are relevant.
The quick stop ramps are set with Object *0x604A/n Velocity quick stop* or with the Parameter *Emergency Stop Clockwise* **424** and *Emergency Stop Anticlockwise* **425**.



If the frequency/speed reaches the value zero during the switch-off time, the drive continues to be supplied with current until the switch-off time has elapsed. This ensures that the drive is at a standstill when the state changes.

10.3.2 Behavior in Transition 5 (Disable Operation)

- **392 state-transition 5**

The behavior in transition 5 (from "Operation released" to "Switched On") can set via parameter *state-transition 5* **392**.

Parameter		Setting		
No.	**Description**	**Min.**	**Max.**	**Factory Setting**
392	state-transition 5	0	2	2

Operation Mode	Function
0 - Coast to Stop	Immediate transition from "Operation released" to "Switched On", drive coasts to a standstill.
1 - DC-Brake	Activation of DC brake, at the end of DC deceleration, there is the change from "Operation released" to "Switched On".
2 - Ramp	Transition at normal ramp, when the drive has come to a standstill, there is the change from "Operation released" to "Switched On".

Note:
Setting operation mode "1 - DC brake" is only possible in applications with v/f characteristic (e.g. configuration 110), as other applications do not know such an operation mode.
If the frequency inverter is operated with a configuration which does not know the DC braking operation mode (e.g. configuration 210, field-orientation speed controlled), value "1" cannot be set. It is also not offered in the selection menus of the Operator Panel or the VPlus PC-Software.

Note:
The default value for parameter *State transition 5* **392** is operation mode "2 - Ramp". For configurations with torque control, the default value is operation mode "0 - Coast to stop".
If the configuration is changed, the value set for *State transition 5* **392** is also altered, if necessary.

If transition 5 has been triggered with *State transition 5* **392** = "1 - DC brake", a new control word is only accepted after the completion of the transition process. The change of state from "Operation enabled" to "Ready" is carried out after the time parameterized for the DC brake *Braking time* **632** has expired.

If the parameter *State transition 5* **392** = "2 - Ramp" has been set, the *controlword* can be set back to "Enable operation" during the stoppage of the drive. In this way, the drive runs back up to its set reference value and remains in the state "Operation enabled".

The change of state from "Operation enabled" to "Switched On" is carried out after the set switch-off threshold has been reached **and** the set holding time (equivalent to the behavior in a quick stop) has expired. In this, the Parameters *Switch-off Threshold Stop Function* **637** (Percentage value of Parameter *Maximum Frequency* **419**) and *Holding Time Stop Function* **638** (holding time after switch-off threshold reached) are relevant.

10.3.3 Reference Value / Actual Value

The PLC gives its reference value to the frequency inverter via Object *0x6042/0 Target Velocity* in the RxPDO used and receives the information on its actual value back via Object *0x6044/0 Control effort* (current Speed) in the TxPDO used.

The use of the reference/actual value channel depends on the set configuration (control system). The actual value is generated from the appropriate source depending on the control system used.

The reference value from Object 0x6042 *Target-velocity* is provided by the Fieldbus Reference Value. In the Frequency Reference Value Channel the setting "20 – Fieldbus Reference Value" can be selected via Parameter *Reference Frequency Source 1* **475** or *Reference Frequency Source 2* **492**.

Actual Values		
Parameter	**Content**	**Format**
Internal reference frequency **228**	Sum of reference values *Reference Frequency Value Source 1* **475** and *Reference Frequency Value Source 2* **492**.	xxx.xx Hz
Reference bus frequency **282**	Fieldbus Reference Value from CANopen® bus	xxx.xx Hz
Reference ramp frequency **283**	Current Reference Frequency of the Ramp	xxx.xx Hz

10.3.4 Example Sequence

One of the following sequences can be used:

1	Control Word =	0x0000	Disable Voltage
2	Control Word =	0x0006	Shutdown
3	Control Word =	0x0007	Switch On
4	Control Word =	0x000F	Enable Operation

OR

1	Control Word =	0x0000	Disable Voltage
2	Control Word =	0x000F	Enable Operation

11 Parameter List

The parameter list is structured according to the menu branches of the Operator Panel. For better clarity, the parameters have been marked with pictograms:

- The parameter is available in the four data sets
- ☑ The parameter value is set by the SET-UP routine
- ⊗ This parameter cannot be written when the frequency inverter is in operation

11.1 Actual Values

No.	Description	Unit	Display Range	Chapter
	Actual Values of the Frequency Inverter			
228	Internal Reference Frequency	Hz	-1000,00 ... 1000,00	10.3.3
249	Active Data Set	-	1 ... 4	10
260	Actual Error	-	0 ... 0xFFFF	9.5.2,12.5
270	Warnings	-	0 ... 0xFFFF	12.3
274	Application Warnings	-	0 ... 0xFFFF	12.4
282	Reference Bus Frequency	Hz	-1000,00 ... 1000,00	10.3.3
283	Reference Ramp Frequency	Hz	-1000,00 ... 1000,00	10.3.3
411	Status Word	-	0 ... 0xFFFF	10
1290	Node-State (NMT)	-	0 ... 127	8.7
1291	CAN-State (physical layer)	-	0 ... 4	5.2
1453	OS SyncSource Act	-	Selection	8.10

Note:
The parameters *Current error* **260**, *Warnings* **270** and *Application Warnings* **270** are only accessible via the manufacturer objects 0x2nnn. They cannot be accessed via the VPlus PC-Software or the Operator Panel.

11.2 Parameters

No.	Description	Unit	Value Range	Fact. Setting	Chapter
	CAN Bus				
276	CAN Interface	-	Selection	1 - CM-CAN	6.4
385	CAN Baud rate	-	Selection	6 - 250 kBit/s	7.1
387	CAN Node Number	-	-1 ... 127	-1	7.2
388	CAN Error behavior	-	Selection	1 - Error	7.3, 9.5.1
	Rated Motor Parameters				
373	No. of Pole Pairs	-	1 ... 24	2	9.5.5
	Bus Control				
392	State transition 5	-	Selection	2 - Ramp	10.3.2
410	Control Word	-	0 ... 0xFFFF	0	10
412	Local/Remote	-	Selection	4 - Ctrl. Key-pad/Contacts	10
	Data Set change-over				
414	Data Set selection	-	0 ... 4	0	10
	Frequency Ramps				
420	Acceleration Clockwise	Hz/s	0,00 ... 9999,99	5,00	9.5.9
421	Deceleration Clockwise	Hz/s	0,01 ... 9999,99	5,00	9.5.10, 9.5.11
422	Acceleration Anticlockwise	Hz/s	-0,01 ... 9999,99	-0,01	9.5.9
423	Deceleration Anticlockwise	Hz/s	-0,01 ... 9999,99	-0,01	9.5.10, 9.5.11

No.	Description	Unit	Value Range	Fact. Setting	Chapter
424	Emergency Stop Clockwise	Hz/s	0,01 ... 9999,99	5,00	9.5.11, 10.3.1
425	Emergency Stop Anticlockwise	Hz/s	0,01 ... 9999,99	5,00	9.5.11, 10.3.1
	Digital Outputs				
549	Reference Value Reached:	%	0,01 ... 20,00	5,00	10.1, 10.2
	Stopping Behavior				
637	Switch-Off Threshold	%	0,0 ... 100,0	1,0	10.3.1, 10.3.2
638	Holding Time Stop Function	s	0,0 ... 200,0	1,0	10.3.1, 10.3.2
	CANopen® Mux/Demux				
1420	CANopen Mux Input Index (write) [1)]	-	EEPROM: 0 ... 16 RAM: 17 ... 33	1	9.4.5
1421	CANopen Mux Input Index (read) [1)]	-	EEPROM: 0 ... 16 RAM: 17 ... 33	1	9.4.5
1422	CANopen Mux Input	-	Selection	7 – Off	9.4.5
	CANopen				
1180	Operation Mode	-	Selection	0 - Off	8.10
1423	CANopen Percentage Actual Value Source	-	Selection	52 – Analog Input MFI1A	9.4.8
1451	CANopen OS Synctime	-	700 ... 900 µs	800 µs	8.10
1452	OS_SyncSource	-	Selection	1 - CANopen	8.10

1)

Non volatile (fixed Parameterization)		Volatile	
0:	All indexes in EEPROM	17:	All indexes in RAM
1...16:	One index in EEPROM	18...33:	One index 1...16 in RAM

Note:
The setting "0" for *CANopen Mux Input Index (write)* **1420** changes all Data in EEPROM and. RAM.

Note:
Parameter *Data Set Selection* **414** is only accessible via the manufacturer objects 0x2nnn. It cannot be accessed via the VPlus PC-Software or the Operator Panel.

12 Appendix

12.1 Control Word Overview

The following table provides an overview of the functions of the bits in the **Control Word** used when the drive is used in Control via Statemachine (*Local/Remote* **412** = "1 - Control via Statemachine").

Bit	*AGL Control word*
0	Switch On
1	Enable Voltage
2	Quick Stop
3	Enable Operation
4	
5	
6	
7	Reset Fault
8	
9	
10	
11	
12	
13	
14	
15	

12.2 Status Word Overview

The following table provides an overview of the functions of the bits in the **Status Word** used when the drive is used in Control via Statemachine (*Local/Remote* **412** = "1 - Control via Statemachine").

Bit	*AGL Status word*
0	Ready to Switch On
1	Switched On
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick Stop
6	Switch On disabled
7	Warning
8	
9	Remote
10	Target reached
11	Internal limit active
12	
13	
14	
15	Warning 2

12.3 Warning Messages

The various control functions and methods as well as the hardware of the frequency inverter contain functions that continuously monitor the application. In addition to the messages documented in the manual, the following warning messages are activated by the CANopen® communication module CM-CAN. The warning messages are given via parameter *Warnings* **270**, bit-coded according to the following scheme. Parameter *Warnings* **269** shows the warnings in clear text on the operator panel and the PC software tool VPlus.
Use Parameter *Warnings* **270** to acces the warning codes via CANopen.

Warning Message		
Bit-No.	**Warning Code**	**Description**
0	0x0001	Warning Ixt
1	0x0002	Warning Short Term - Ixt
2	0x0004	Warning Long Term - Ixt
3	0x0008	Warning Heat Sink Temperature Tk
4	0x0010	Warning Inside Temperature Ti
5	0x0020	Warning Limit
6	0x0040	Warning Init
7	0x0080	Warning Motor Temperature
8	0x0100	Warning Mains Failure
9	0x0200	Warning Motor Protective Switch
10	0x0400	Warning Fmax
11	0x0800	Warning Analog Input MFI1A
12	0x1000	Warning Analog Input MFI2A
13	0x2000	Warning Systembus
14	0x4000	Warning Udc
15	0x8000	Warning *Application Warnings* **274**

Note:
The meaning of the individual warnings is described in detail in the operating instructions.

12.4 Warning Messages Application

When the highest bit in the Warning messages is set, a "Warning Message Application" is present. The Application warning messages are given via parameter *Application Warnings* **274**, bit-coded according to the following scheme.
Parameter *Application Warnings* **273** shows the warnings in clear text on the operator panel and the PC software tool VPlus.
Use Parameter *Application Warnings* **274** to access the Application warning codes via CANopen.

Warning Messages		
Bit-No.	**Warning Code**	**Description**
0	0x0001	BELT
1	0x0002	(reserved)
2	0x0004	(reserved)
3	0x0008	(reserved)
4	0x0010	(reserved)
5	0x0020	(reserved)
6	0x0040	SERVICE
7	0x0080	User 1
8	0x0100	User 2
9	0x0200	(reserved)
10	0x0400	(reserved)
11	0x0800	(reserved)
12	0x1000	(reserved)
13	0x2000	(reserved)
14	0x4000	(reserved)
15	0x8000	(reserved)

Note:
The meaning of the individual Application Warnings is described in detail in the operating instructions.

12.5 Error Messages

The error code that is stored after an error occurs is made up of the error group FXX (high-Byte, hexadecimal) followed by the code number XX (low-Byte, hexadecimal).

Communication Error			
Code		**Meaning**	
F20	21	CAN Bus-OFF	
	22	CAN Guarding	
	23	Error state	
	24	SYNC error (SYNC timing)	
	25	CAN NMT Error	
	26	RxPDO1 length error	(no. of received bytes different from the mapping)
	27	RxPDO2 length error	
	28	RxPDO3 length error	
F23	nn	CAN Heartbeat, nn = Node-ID of the failed node (hex)	

The Actual error message can also be read out by parameter access via parameter *Actual Fault* **260**. Parameter *Actual Error* **259** shows the actual error in clear text on the operator panel and the PC software tool VPlus.

In addition to the error messages listed, there are further error messages used for internal purposes only, which are not listed here. If you receive any error messages which are not listed, please contact us.

Index

Bonfiglioli

power, control and green solutions

Bonfiglioli has been designing and developing innovative and reliable power transmission and control solutions for industry, mobile machinery and renewable energy applications since 1956.

www.bonfiglioli.com

Bonfiglioli Riduttori S.p.A.
Via Giovanni XXIII, 7/A
40012 Lippo di Calderara di Reno
Bologna, Italy

tel: +39 051 647 3111
fax: +39 051 647 3126
bonfiglioli@bonfiglioli.com
www.bonfiglioli.com

VEC 676 R0