Bonfiglioli
Vectron

Active Cube

MODBUS
Communication Module
CM-MODBUS / CM2-MODBUS
Frequency Inverter 230 V / 400 V

Bonfiglioli

1 General Information about the Documentation

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

The present manual was created in the German language. The German manual is the original version. Other language versions are translations.

Quick Start Guide

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

User manual

The user manual documents the complete functionality of the frequency inverter. The parameters required for special purposes, for adjustment to the application and the numerous additional functions are described in detail.

Separate user manuals are supplied for optional components for the frequency inverter. These manuals complement the operating instructions and the "Quick Start Guide" for the frequency inverter.

Application manual

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

Installation instructions

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

1.1 This document

The present user manual of the CM-MODBUS and CM2-MODBUS communication module complements the Operating Instructions and the "Quick Start Guide" for the frequency inverters of the ACU 201, ACU 401, ACU210 und ACU410 device series.

The user manual contains important information on the installation and use of the MODBUS communication module CM-MODBUS in its specified application range. Compliance with this user manual contributes to avoiding risks, minimizing repair cost and downtimes and increasing the reliability and service live of the frequency inverter.

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

⚠ **WARNING**

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

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

In this manual, the designation „CM-MODBUS" also applies to CM2-MODBUS modules.

1.2 Warranty and liability

BONFIGLIOLI VECTRON GmbH would like to point out that the contents of this user manual do not form part of any previous or existing agreement, assurance or legal relationship. Neither are they intended to supplement or replace such agreements, assurances or legal relationships. Any obligations of the manufacturer shall solely be based on the relevant purchase agreement which also includes the complete and solely valid warranty stipulations. These contractual warranty provisions are neither extended nor limited by the specifications contained in this documentation.

The manufacturer reserves the right to correct or amend the specifications, product information and omissions in these operating instructions without notice. The manufacturer shall not be liable for any damage, injuries or costs which may be caused by the aforementioned reasons.

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

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

1.3 Obligation

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

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

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

1.4 Copyright

In accordance with applicable law against unfair competition, this user manual is a certificate. Any copyrights relating to it shall remain with

BONFIGLIOLI VECTRON GmbH
Europark Fichtenhain B6
47807 Krefeld
Germany

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

Any non-compliance will constitute an offense against the copyright law dated 09 September 1965, the law against unfair competition and the Civil Code and may result in claims for damages. All rights relating to patent, utility model or design registration reserved.

1.5 Storage

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

1.6 Final decommissioning

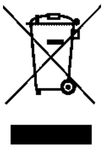
After the end of product service life, the user/operator must take the device out of operation.

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

Disposal requirements under European Union WEEE regulations

The product is marked with the WEEE symbol shown below.

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



2 General safety instructions and information on use

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

Terminology

According to the documentation, different activities must be performed by certain persons with certain qualifications.

The groups of persons with the required qualification are defined as follows:

Operator

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

Operating staff

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

Skilled Personnel The term **Skilled Personnel** covers staff that are assigned special tasks by the Operator of the frequency inverter, e.g. installation, maintenance and service/repair and troubleshooting. Based on their qualification and/or know-how, **Skilled Personnel** must be capable of identifying defects and assessing functions.

Qualified electrician

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

Instructed person

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

Expert

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

2.1 Designated use

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

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

The frequency inverters meet the requirements of the low voltage directive 2014/35/EU and DIN EN 61800-5-1. CE-labelling is based on these standards. Responsibility for compliance with the EMC Directive 2014/30/EU lies with the operator. Frequency inverters are only available at specialized dealers and are exclusively intended for commercial use as per EN 61000-3-2.

No capacitive loads may be connected to the frequency inverter.

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

2.2 Misuse

Any use other than that described in "Designated use" shall not be permissible and shall be considered as misuse.

For example, the machine/plant must not be operated

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

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

2.3 Explosion protection

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

2.4 Residual risks

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

Typical residual hazards include:

Electrical hazard

Danger of contact with energized components due to a defect, opened covers or enclosures or improper working on electrical equipment.

Danger of contact with energized components inside of the frequency inverter if no external disconnection device was installed by the operator.

Electrostatic charging

Touching electronic components bears the risk of electrostatic discharges.

Thermal hazards

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

Charged capacitors in DC link

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

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

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

2.5 Safety and warning signs on the frequency inverter

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

2.6 Transport and storage

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

The units may only be stored in dry rooms which are protected against dust and moisture and are exposed to small temperature deviations only. The requirements of DIN EN 60721-3-1 for storage, DIN EN 60721-3-2 for transport and labeling on the packaging must be met.

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

2.7 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.

2.8 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.[1)]
- 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.

2.9 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.

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.

2.10 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.

2.11 Disposal

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

3 Introduction

This document describes the Modbus protocol of the communication modules CM-232 with RS232 port and CM-485 with RS485 port. This protocol can be used alternatively to the VA-Bus via the same hardware (CM-232/CM-485). Both protocols can not be operated at the same time. The Modbus operation modes RTU and ASCII are available.

For the RS232 connection, the frequency inverter must be equipped with the RS232 communication module CM-232.

For the RS485 connection, the frequency inverter must be equipped with the RS485 communication module CM-485.

The communication modules CM-232 and CM-485 are separate components and must be mounted to the frequency inverter by the user. This is described in detail in chapter "Assembly".

Note:

This document only describes the communication modules CM-232 and CM-485. It does not provide basic information on the serial interfaces RS232 and RS485. Neither does it contain any basic information on the operation of the frequency inverters.

Basic knowledge of the methods and function of the RS232 and RS485 serial interfaces and Modbus protocol is required in order to understand and use the instructions contained in this document.

Note:

In some chapters of these instructions, setting and display options via the PC software VPlus are described as an alternative to the KP500 control unit. In this case, VPlus communicates with the frequency inverter

- via the module CM-232 or CM-485 **or**
- via the interface adapter KP232.

If the serial interface of the module CM-232/CM-485 is connected to a PLC, for example, VPlus cannot access the frequency inverter via this interface directly.

In this case, the PC must be connected via the optional interface adapter KP232.

Attention!

Via the RS232 port at CM-232 or the RS485 port at CM-485, a controller can access **all** parameters of the frequency inverter.

Any change of parameters, the meaning of which is unknown to the user, may result in the frequency inverter becoming inoperative.

RS485 connection:

Frequency inverters can be connected to a bus system by extension with CM-485 communication modules. The bus structure is linear and designed as a 2-wire line. Via a bus master, up to 247 frequency inverters can be addressed.

The frequency inverters can be parameterized and controlled easily. Additionally, frequency inverter data can be polled and positioning commands can be transmitted during operation via the bus system by means of a PC or a PLC.

RS232 connection:

The RS232 connection enables a peer-to-peer connection between two clients.

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!

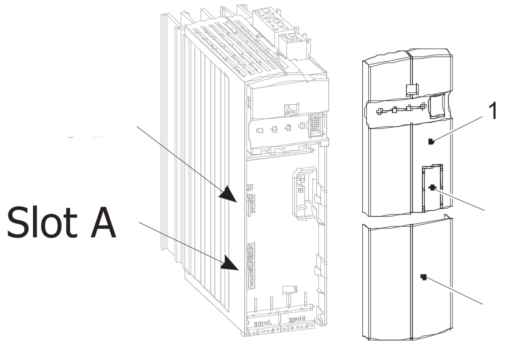
The frequency inverter must be disconnected from the power supply before installation of the communication module.

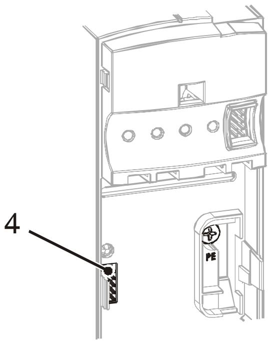
Installation under voltage is not permitted and will destroy the frequency inverter and/or the communication module.

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

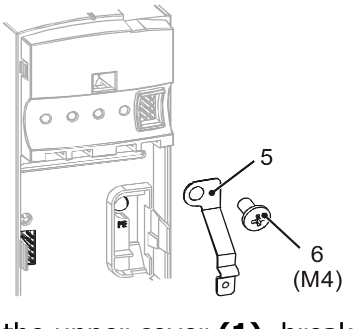
Work steps:

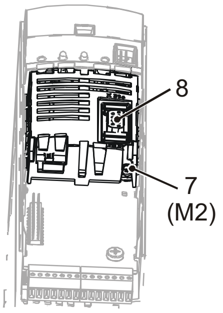
- Disconnect the frequency inverter from mains voltage and protect it against being energized unintentionally.
- Remove covers **(1)** and **(2)** of the frequency inverter. Slot B **(4)** for the communication module is now accessible.



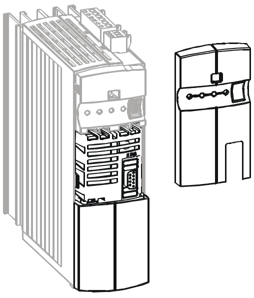


- Mount the supplied PE spring **(5)** using the M4 screw **(6)** in the unit. The spring must be aligned centrally.
- Insert the communication module in slot B **(4)** until it engages audibly.
- Fix the communication module by screwing the M2 screw **(7)** of the module to the PE spring **(5)**.



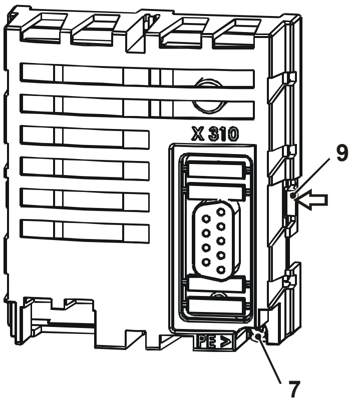


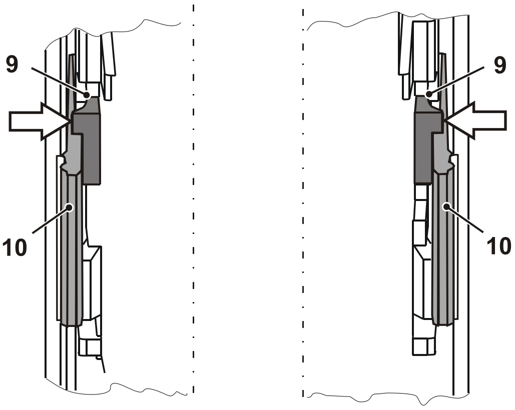
- In the upper cover **(1)**, break out the pre-punched cutout **(3)** for the plug X310 **(8)**.
- Mount the two covers **(1)** and **(2)**.



4.2 Disassembly

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





- Loosen the M2 screw **(7)** at the communication module

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

The looking hooks **(9)** are located at the place where the looking hooks **(10)** for the upper cover **(1)** project from the case of the frequency inverter.

- To do this, insert the screwdriver in the gap between the case of the module and the frequency inverter carefully and push the locking hook inwards in the direction of the arrow **(⇦)**. As soon as the right side is unlocked, pull out the module a bit on the right side and hold it.
- Hold the module on the right side while unlocking the locking hook on the left side in the same way **(⇨)**.
- Pull the module out of the slot by gently pulling on the right and left side alternately.
- Disassemble the PE spring (5).
- Mount the two covers (1) and (2).

5 RS485 and RS232 Interfaces

The frequency inverter can be controlled from a PLC or another master device via a serial interface using the Modbus or VABus protocol. The VABus protocol is required for the parameterization with the VPlus PC-Software.

The connection can be established via theoption KP232 or via an optional Communication Module.

The following protocols can be selected:

	KP232	CM485 and CM232
Modbus RTU		X
Modbus ASCII		X
VABus	X	X

The communication with the Modbus-Protocol can be established via:

- the option KP232 with RS232-connection using a 9-pin D-Sub connector
- the CM-232 Communication Module with RS232-connection using a 9-pin D-Sub connector
- the CM-485 Communication Module with RS485-connection using a 9-pin D-Sub connector
- the CM-485T Communication Module with RS485-connection using a 7-pin terminal socket

The Modbus communication is possible either via CM485 (serial Bus) or via CM232 (serial point to point connection).

VABus communication is always possible via option KP232 and optional via a communication module. KP232 and a communication module can be used with VABus protocol at the same time.

Possible Combinations of VABus with Modbus:

Communication Module		X21 (RJ45)
Modbus (RTU or ASCII)	and	VABus
VABus	and	VABus

Combination Options with the Scope Function:

Communication Module		X21 (RJ45)
VABus	and	Scope Function (VABus)
Scope Function (VABus)	and	VABus
Modbus (RTU or ASCII)	and	Scope Function (VABus)

The Scope Function is started via the VPlus PC-Software. The Scope Function cannot be started via VPlus and an optional Communication Module at the same time.

The baud rates for the KP232 is detected automatically, the baud rate of the CM232 or CM485 module is set up via parameter.

Note: This document is not basic information for the RS232 or RS485 serial interface. Fundamental knowledge of the Modbus protocol and the RS232 and RS485 serial interfaces is a prerequisite.

In some sections – as an alternative to control via the operator panel – settings and display possibilities are described with the help of the VPlus PC-Software. Here, VPlus communicates with the frequency inverter via the X21-Connection or via an optional CM-232 or CM-485 Communication Module.

If the serial interface of an optional CM-232 or CM-485 Communication Module is connected to a PLC, then simultaneous access to the frequency inverter from the VPlus PC-Software is no longer possible. In this case the connection to the PC can be made via USB with the help of an optional interface adapter on the X21-Connection.

Warning!

Via the Modbus-Communication, a control unit can access all of the frequency inverter parameters.

The changing of parameters, whose meaning is not known to the user, can lead to the malfunctioning of the frequency inverter and to dangerous situations in the plant.

Caution!

If values are to be written regularly with a high repetition rate, then no entry may be made to the EEPROM, as it only has a limited number of admissible write cycles (approx. 1 million cycles). If the number of allowed write cycles is exceeded then the EEPROM will be damaged. See Chapter 8 "Handling of Data Sets / Cyclic Writing".

RS485-Connection

Frequency inverters can be connected to a bus system using CM-485 Communication Modules. The bus structure is linear and implemented as a 2-wire line. Up to 247 frequency inverters can be addressed and polled from a bus master via Modbus.

The frequency inverters can be parameterized and controlled via the bus system. During operation data can be requested and set from a PC or PLC.

RS232-Connection

The RS232-Connection allows a point to point connection between the participants.

5.1 Communication Modules

Direct connection of the CM-232 to a PC or PLC

The CM-232 Communication Module enables a direct connection between a 9-pin D-Sub connector (X310) of the CM-232 and the serial interface of a PC or PLC.

The configuration of the installed communication module is carried out using the VPlus PC-Software or using the Operator Panel.

With CM-232 the connection to the PC or PLC is made via a RS232-Connection Line (1:1 occupancy).

Direct connection of the CM-485 to a PC or PLC

For the direct connection between the 9-pin D-Sub Interface (X310) of the CM-485 and the serial RS232-Interface of a PC or PLC, install an RS485/RS232 interface adapter on the signal line.

The configuration of the installed communication module is carried out using the VPlus PC-Software or using the Operator Panel.

Attention!

The transmitter and receiver must be set to the same transfer rate (baud rate).

The set baud rate applies to the CM-232 and CM-485 communication modules.

Work Steps:

- Mount the CM-232 / CM-485 Communication Module onto the frequency inverter.
- For the CM-232 Communication Module:
  Connect the CM-232 and PC with an RS232-Cable.
- For the CM-485 Communication Module:
  Connect the CM-485 with a RS485-Cable to the RS485/RS232 Interface Adapter.
  Connect the RS232-Connection of the Interface Adapter with the PC/PLC.

Setting the type of Protocol

The factory setting of the CM-232/CM-485 Communication Modules is the BONFIGLIOLI VECTRON standard protocol (VABus). Communication with the VPlus PC-Software is only possible using this protocol.

The Modbus protocol enables the operation of a straight Master/Slave-System. The Bus-Master can be a PC, a PLC or any arbitrary computer system.

Note:

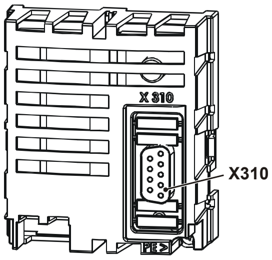
The option KP232 always uses the protocol VABus independent of the setting of the communication module.

5.1.1 Installation Notes

- For the RS485 bus cable use a twisted, shielded cable.
- Implement the shield as a braided shield (not a foil shield).
- Connect the cable shield surfaces at both ends to PE.
- The connector pin assignments of an RS485-Cable and an RS232-Cable are different. No data transfer is possible if the wrong cable is used.

5.1.2 Pin Assignment

5.1.2.1 RS232 Communication Module CM-232 DB9



The RS232-Interface is connected to a PC or a controller via the 9-pin D-Sub socket X310.

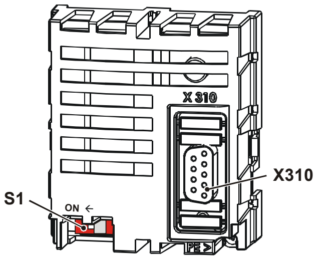
The assignment complies with the standard, so that only an RS232 connection cable (1:1) is required.

Bus Connector X310 CM-232 (9-pin D-Sub)

Pin	Name	Function
Housing	Shield	connected with PE
1	–	n. c.
2	RxD	receive data (input)
3	TxD	transmit data (output)
4	–	n. c.
5	0 V	Ground
6	–	n. c.
7	–	n. c.
8	–	n. c.
9	–	n. c.

n.c: not connected

5.1.2.2 RS485 Communication Module CM-485 DB9



The RS232-Interface is connected to a PC or a controller via the 9-pin D-Sub socket X310.

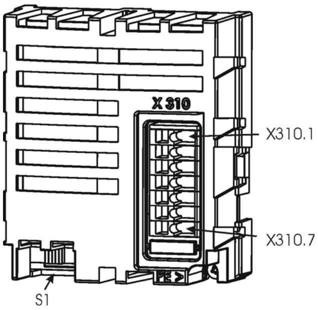
For details on the pin assignment, refer to the following table.

Bus Connector X310 CM-485 (9-pin D-Sub)

Pin	Name	Function
Housing	Shield	connected with PE
1	Data Line B	short-circuit proof and functionally insulated; max. current 60 mA
2	Data Line B’	same as pin 1 – for cable network
3	0 V	GND/earth
4	–	n. c.
5	+5 V	Interface converter supply voltage +5 V
6	–	n. c.
7	Data Line A	short-circuit proof and functionally insulated; max. current 60 mA
8	Data Line A’	same as pin 7 – for cable network
9	–	n. c.

n.c: not connected

5.1.2.3 RS485 Communication Module CM-485T



The RS485-Interface is connected via the 7-pin terminal socket X310.
For details on the pin assignment, refer to the following table.

Bus Connector X310 CM-485 T (7-pin terminal socket)

Terminal	Name	Function
1	Data Line A	short-circuit proof and functionally insulated; max. current 60 mA
2	Data Line A’	same as pin 1 – for cable network
3	Data Line B	short-circuit proof and functionally insulated; max. current 60 mA
4	Date Line B’	same as pin 3 – for cable network
5	+5 V	Interface converter supply voltage +5 V
6	0 V	GND/earth
7	PE	Shield

5.1.3 RS485 Bus Termination

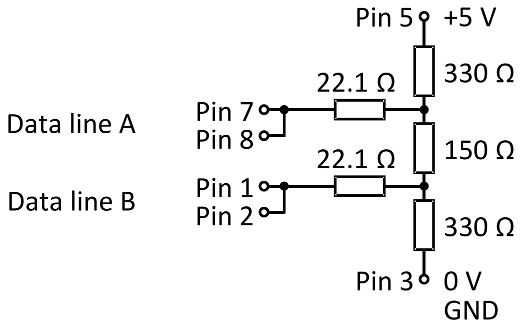
Attention!

The passive bus termination (connection of a termination resistor), is required at the physically first and last client. It can be activated by the CM-485 and CM-485T DIP Switch S1.

By default, the bus termination is set to OFF.

It is important to implement a correct termination. Otherwise, no communication is possible via the RS485-Interface.

As an alternative, the active bus termination is possible via a corresponding circuit:

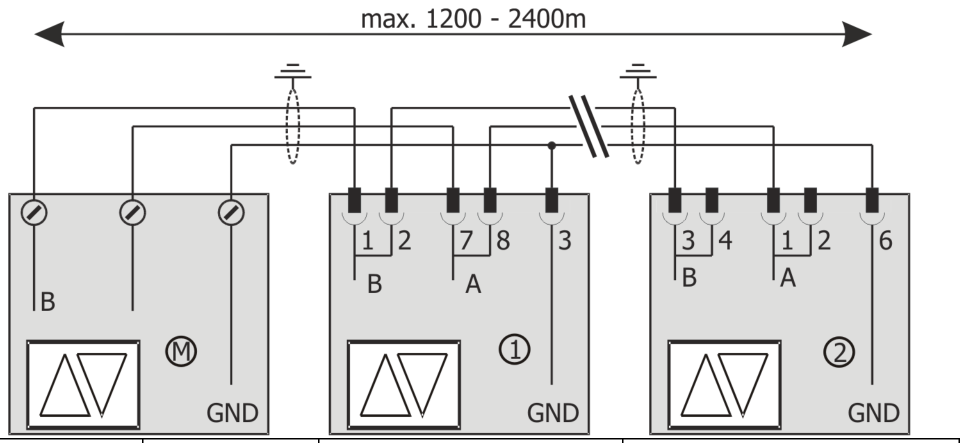


The active termination is only allowed once on each branch. The bus termination via an external circuit and via DIP switch at the same time is not allowed.

Pay attention to the ground wiring. This will protect the communication bus against high noise level.

For easy wiring the signal terminals A and B have parallel contacts.

Example of wiring with different CM-485 modules:



M	Master (i.e. PC)			Termination ON
1	Inverter 1	CM-485 DB9	9-pin D-Sub	Termination OFF
2	Inverter 2	CM-485T	7-pin terminal socket	Termination ON

5.2 Commissioning

By default, the parameters of the communication modules CM-232 and CM-485 are set as follows:

Parameter		Settings
No.	Description	Factory setting
395	Protocol Type	0 - VABus
413	Watchdog Timer	0

Parameter		Settings
No.	Description	Factory setting
10	Baud Rate	4 - 19200

1375	Modbus Parity	0 – EVEN
1376	Modbus Address	1

Set up the parameters for using Modbus protocol according to the application.

5.2.1 Setting the Modbus Baud Rate

The transmission speed and parity of the CM-232 are set via parameters *Baud Rate* **10** and *Modbus Parity* **1375**.

The transmission speed of the CM-232 depends on various application-specific parameters. For example, the cable length limits the transmission speed due to signal propagation delays.

Baud Rate 10		Function	max. cable length [1]	
			CM-232	CM-485
1 –	2400 Baud	Transmission rate 2400 Baud	30 m	2400 m
2 –	4800 Baud	Transmission rate 4800 Baud	30 m	2400 m
3 –	9600 Baud	Transmission rate 9600 Baud	30 m	1200 m
4 –	19200 Baud	Transmission rate 19200 Baud	30 m	1200 m
5 –	57600 Baud	Transmission rate 57600 Baud	10 m	600 m
6 –	115200 Baud	Transmission rate 115200 Baud	10 m	300 m

[1] The indicated cable lengths are recommended maximum values which depend, among other things, on the quality of the cable.

Attention:

- Baud rate and parity changes are only active after a reset of the frequency inverter via the software or Mains Off/On.
- In the case of the software reset, proceed as follows:
- Via the control unit KP500 or the control software VPlus open parameter *Program(ming)* **34**.
- Set parameter value "123".
- Confirm by pressing "ENT".
- After the reset, the frequency inverter is initialized and is ready for operation after a few seconds.

5.2.2 Setting the Modbus Parity

With Parameter *Modbus Parity* **1375** the parity of the Modbus communication can be set.

Modbus Parity **1375**		Function
0 –	Even	Even parity is used in the data transfer.
1 –	Odd	Odd parity is used in the data transfer.
2 –	None	No parity is used in the data transfer.

Attention!
Changes are effective immediately and without a restart of the frequency inverter.

5.2.3 Setting the Modbus Address

The node address can be set with Parameter *Modbus Address* **1376**. Up to 247 frequency inverters can be operated with Modbus. These are allocated a unique address in the range 1 to 247.

Parameter		Setting		
Nr.	**Description**	**Min.**	**Max.**	**Factory Setting**
1376	Modbus Address	1	247	1

When using a CM232 module (point to point connection) in most cases a change of the factory setting is not necessary (address 1). For certain operating conditions the addressing can be adjusted.

Attention!

Changes are effective immediately and without a restart of the frequency inverter.

An address must be allocated for each participant.

Double assignments are not permitted when allocating bus addresses.

5.2.4 Setting the Protocol

The Modbus protocol described in the following has been incorporated into the standard ACT/ACU software of BONFIGLIOLI VECTRON. It defines and describes the communication via the serial interfaces RS232/RS485 using the Modbus RTU and ASCII transmission modes.

By default, the frequency inverters are set to the standard VABus protocol. The Modbus protocol type can be set via parameter *Protocol* **395**:

Protocol 395		Function
0 -	VABus	BONFIGLIOLI VECTRON standard protocol (default setting)
1 -	P-Bus	Application-specific bus protocol [1]
2 -	Modbus RTU	Modbus over serial line (with RTU Transmission Mode)
3 -	Modbus ASCII	Modbus over serial line (with ASCII Transmission Mode)

Note:

[1] This bus protocol is only used for special applications. It cannot be used for standard applications.
For more detailed information, refer to the application-specific instructions.

Attention:

Changes of the parameter *Protocol* **395** are effective immediately, i.e. without a restart of the frequency inverter.

If the wrong protocol is selected, communication via CM-232/CM-485 is not possible. In this case, correct the protocol type using the control unit KP500 or KP232 and VPlus.

The Modbus protocol enables the operation as a pure Master/Slave system. A PC, a PLC or any type of computer system is the bus master.

5.2.5 Setting the Watchdog Timer

The communication can be monitored. If the communication fails, then no data or incorrect data will be transferred. This state will be detected by the Communication Watchdog.

The Watchdog-Function monitors the time, within which no correct communication occurs. This time can be set with Parameter *RS232/RS485 Watchdog Timer* **413**. The set value is the time in seconds within which at least one correct data exchange must occur.

If the adjusted monitoring time has elapsed, the frequency inverter signals a fault. The fault message **F2010 Watchdog RS232/RS485** is displayed.

Parameter		Setting		
Nr.	**Description**	**Min.**	**Max.**	**Factory Setting**
413	RS232/RS485 Watchdog Timer	0 s	10000 s	0 s

If the parameter is set to zero (factory setting), then monitoring is switched off.

6 Protocol

The MODBUS serial communication protocol is a Master/Slave-Protocol. With the bus only one master is connected (at a time). One or several (max. 247) slave-nodes are connected with each other on the same bus. A MODBUS communication is always initiated by the master. The slave nodes do not communicate with each other. The master only initiates one MODBUS communication at a time.

The master sends a MODBUS request to the slave-nodes in two different operation modes::

- In the **Unicast** operation mode the master addresses a single slave-node. After reception and processing of the request, the slave-node sends a message (reply) to the master. In this operation mode a MODBUS transfer consists of two messages: a request from the master and a reply from the slave-node. Each slave-node must have a unique address (from 1 to 247), so that the slave-nodes can receive requests independent of each other.
- In the **Broadcast** operation mode the master can send a request to all slave-nodes. The slave-nodes do not send a reply to the request.. The request consists of write commands. All slaves must accept the write commands. Address 0 is reserved for the recognition of a broadcast request.

6.1 Address Representation

Up to 247 frequency inverters can be operated on MODBUS. These are assigned addresses in the range 1 … 247. With address 0, all clients connected to the bus can be addressed simultaneously. Address 0 is also referred to as the Broadcast-Address.

6.2 Message Structure

A MODBUS message is made up of the following fields:

Address	Function Code	Data	CRC (or LRC)

The **Address field** contains the slave-address in the Modbus message.

Valid slave node addresses are in the range of 0 – 247 (decimal). The individual slave devices are assigned addresses in the range of 1 – 247. A master addresses a slave by placing the slave address in the address field of the message. When the slave returns its response, it places its own address in the response address field to let the master know which slave is responding.

The **Function Code** indicates to the inverter what kind of action to perform. The function code can be followed by a *Data* field that contains request parameters (or, in the case of the inverter's response, the response parameters).

If no error occurs related to a correctly received Modbus request then the data field contains the data requested. If an error occurs then the field contains an Exception Code to indicate to the master that the request was unsuccessful. Exception Responses and Codes are described in detail in chapter 7.3.9 "Exception Codes".

If a request is not received correctly (communication error, checksum error) then no response is sent. The inverter waits for the next request.

The **Error Checking Field** is the result of a "Redundancy Checking" calculation that is performed on the message contents. Two kinds of calculation methods are used depending on the transmission mode that is being used (RTU or ASCII). See Chapter 7.3.10 "Modbus Transmission Modes".

6.3 Supported Function Codes

Modbus definitions for reading/writing data in a device do not fit directly to parameter access of inverters (independent of inverter manufacturer). Modbus is defined for reading/writing bits and registers in a different way. Furthermore, data access is limited to 16 bit wide data.

To fulfill the Modbus requirements the data access to parameters in the devices (inverters) will use the following defined function codes:

16 bits values:

- Function Code 3, Read ONE 16 bit wide data *(Read Holding Register)*
- Function Code 6, Write ONE 16 bit wide data *(Preset Single Register)*
- Function Code 16, Write ONE 16 bit wide data *(Preset Multiple Register)*
-

32 bits values:

For data access to 32 bit wide data two new inverter specific function codes are defined:

- Function Code 3, Read two 16 bit (=32 Bit) wide data *(Read Holding Register)*
- Function Code 16, Write two 16 bit (=32 Bit) wide data *(Preset Multiple Register)*
- Function Code 100, Read ONE 32 bit wide data
- Function Code 101, Write ONE 32 bit wide data

Note:

The Modbus specification doesn't specify the handling of 32 bit values. The implemented handlings and function codes to access 32 Bit values are however widely spread and commonly used. These functions allow data access to 32-bit "long" variables/parameters in the inverter.

Note:

The Function Codes 3 and 16 for 32 Bit access are supported in ACU starting with 5.1.12. For ACT the support of the function Codes 3 and 16 for 32 Bit access is in preperation.

Diagnosis:

For diagnostic purposes the Modbus function code "8" (*Diagnostics*) is also supported.

These function codes and their corresponding data fields are described in detail in the following chapters.

Note:

In all data fields with more than one byte, the **highest** order byte is transmitted first.

Message examples for all supported function codes in both RTU and ASCII transmission modes can be found in Chapter 9 "Modbus Message Examples".

6.3.1 Function Code 3, Read 16-Bit or 32-Bit-Parameter

This function code is used to read the value of 16 Bit or 32 Bit parameters in the inverter.

Request 16 Bit parameter read:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x03
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0001
CRC or LRC & Endframe		

Response 16 Bit parameter read:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x03
No. of Bytes	1 Byte	0x02
Register Value (Parameter Value)	2 Bytes	0 – 0xFFFF
CRC or LRC & Endframe		

Request 32 Bit parameter read:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x03
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0002

CRC or LRC & Endframe		

Response 32 Bit parameter read:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x03
No. of Bytes	1 Byte	0x04
Register Value (Parameter Value)	4 Bytes	0 – 0xFFFFFFFF
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0x83
Exception Code	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Note:

The Function Codes 3 and 16 for 32 Bit access are supported in ACU starting with 5.1.12. For ACT the support of the function Codes 3 and 16 for 32 Bit access is in preperation.

Start Address

This field is used to store the parameter number and data set number. The parameter number ranges from 0 - 1599 and is stored in the lower 12bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

Example: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

No. of Registers

This field is used to store the no. of parameters to be written. The value must always be 1 as it is only possible to write one parameter at a time.

Byte Count

This field is set to

- 2 for 16 Bit parameters.
- 4 for 32 Bit parameters.

Register Value

This field is contains the 16-bit or 32-bit parameter value.

Note:

Parameter values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example:

A current value of 10.3A is transmitted. The numerical value actually transmitted is 103, which corresponds to 0x67 in HEX format.

Exception Code

The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	– No. of Registers field not equal to 1. – Parameter unknown
3	ILLEGAL DATA VALUE	– No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	– Error on reading the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Examples:

	16 Bit	32 Bit
Modbus RTU	9.1.1.1	9.1.2.1
Modbus ASCII	9.2.1.1	9.2.2.1

6.3.2 Function Code 6, Write 16-Bit-Parameter

This function code is used to write the value of an integer or unsigned integer parameter in the inverter.

Request 16 Bit parameter write:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x06
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
Register Value (Parameter Value)	2 Bytes	0 – 0xFFFF
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x06
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
Register Value (Parameter Value)	2 Bytes	0 – 0xFFFF
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0x86
Exception Code	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Start Address

This field is used to store the parameter number and data set number. The parameter number ranges from 0 – 1599 and is stored in the lower 12 bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

e.g.: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

Register Value

This field is used to store the 16-bit parameter value.

Note:

Parameter values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example:

A current value of 10.3A is to be transmitted. The numerical value actually transmitted is 103, which corresponds to 0x67 in HEX format.

Exception Code

The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	–	Parameter unknown
3	ILLEGAL DATA VALUE	–	No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	–	Error on writing the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Example of a Modbus ASCII telegram: see chapter 9.2.1.2.

Example of a Modbus RTU telegram: see chapter 9.1.1.2.

6.3.3 Function Code 16, Write 16-Bit-Parameter

Function code 16 can be used to to write the value of 16 Bit parameter in the inverter.

Request 16 Bit parameter write:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x10
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0001
No. of Bytes	1 Byte	0x02
Register Value (Parameter Value)	2 Bytes	0 – 0xFFFF
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x10
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0001
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0x90
Exception Code	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Start Address

This field is used to store the parameter number and data set number. The parameter number ranges from 0 – 1599 and is stored in the lower 12 bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

e.g.: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

Register Value

This field is used to store the 16-bit parameter value.

Note:

Parameter values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example:

A current value of 10.3A is to be transmitted. The numerical value actually transmitted is 103, which corresponds to 0x67 in HEX format.

Exception Code

The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	–	Parameter unknown
3	ILLEGAL DATA VALUE	–	No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	–	Error on writing the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Example of a Modbus ASCII telegram: see chapter 9.2.1.3

Example of a Modbus RTU telegram: see chapter 9.1.1.3

6.3.4 Function Code 16, Write 32-Bit-Parameter

Function code 16 can be used to to write the value of 16 Bit parameter in the inverter.

Request 32 Bit parameter write:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x10
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0002
No. of Bytes	1 Byte	0x04
Register Value (Parameter Value)	2 Bytes	0 – 0xFFFF FFFF
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x10
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
No. of Registers	2 Bytes	0x0002
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0x90
Exception Code	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Start Address

This field is used to store the parameter number and data set number. The parameter number ranges from 0 – 1599 and is stored in the lower 12 bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

e.g.: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

Register Value

This field is used to store the 32-bit parameter value.

Note:

Parameter values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example:
A frequency value of 123.45 Hz is to be transmitted. The numerical value actually transmitted is 12345, which corresponds to 0x3039 in HEX format.

Exception Code
The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	–	Parameter unknown
3	ILLEGAL DATA VALUE	–	No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	–	Error on writing the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Example of a Modbus ASCII telegram: see chapter 9.2.2.2.

Example of a Modbus RTU telegram: see chapter 9.1.2.2.

6.3.5 Function Code 100, Read 32-Bit-Parameter

Request:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x64
Start Address (Data Set / Para-No.)	2 Bytes	0x0000 – 0x963F
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x64
Register Value (Parameter Value)	4 Bytes	0 – 0xFFFF FFFF
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0xE4
Exception Code	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Start Address
This field is used to store the parameter number and data set number. The parameter number ranges from 0 – 1599 and is stored in the lower 12 bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

e.g.: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

No. of Registers
This field is used to store the 32-bit parameter value.

Note:
Parameter Values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example:
A frequency value of 100.25 Hz is to be transmitted. The numerical value actually transmitted is 10025, which corresponds to 0x2729 in HEX format.

Exception Code
The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	–	Parameter unknown
3	ILLEGAL DATA VALUE	–	No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	–	Error on reading the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Example of a Modbus ASCII telegram see chapter 9.2.2.2.

Example of a Modbus RTU telegram see chapter 9.1.2.39.1.2.3.

6.3.6 Function Code 101, Write 32-Bit-Parameter

Request:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x65
Start Address (Data Set/Para-No.)	2 Bytes	0x0000 – 0x963F
Register Value (Parameter Value)	4 Bytes	0 – 0xFFFF FFFF
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Function Code	1 Byte	0x65
Start Address (Data Set/Para-No.)	2 Bytes	0x0000 – 0x963F
Register Value (Parameter Value)	4 Bytes	0 – 0xFFFF FFFF
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address	1 Byte	1 – 0xF7 (=247)
Error Code	1 Byte	0xE5
Exception Response	1 Byte	2, 3 or 4
CRC or LRC & Endframe		

Start Address
This field is used to store the parameter number and data set number. The parameter number ranges from 0 – 1599 and is stored in the lower 12 bits; the data set number ranges from 0 – 9 and is stored in the upper 4 bits.

e.g.: Parameter **372** (hex. 0x174), data set 2 (hex. 0x2) is stored as hex. 0x2174.

	Start Address															
	Data Set				Parameter number											
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	For the above example:															
Hex.	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	0
Bin.	2				1				7				4			

Register Value
This field is used to store the 32-bit parameter value.

Note:
Parameter values with decimal places are transferred without the decimal point. Depending on the number of decimal places, the values are multiplied by 10, 100 or 1000.

Example: Frequency Value

A frequency value of 100.25 Hz is to be transmitted. The numerical value actually transmitted is 10025, which corresponds to 0x2729 in HEX format.

Exception Code

The following exception codes can occur:

2	ILLEGAL DATA ADDRESS	–	Parameter unknown
3	ILLEGAL DATA VALUE	–	No. of bytes in the data field too small or too large
4	SLAVE DEVICE FAILURE	–	Error on writing the parameter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Example of a Modbus ASCII telegram: see chapter 9.2.2.4.

Example of a Modbus RTU telegram: see chapter 9.1.2.4.

6.3.7 Function Code 8, Diagnostics

This function code is used to access the Modbus diagnostic counters that are supported by the inverter. Each counter can be obtained by a sub-function code bound to the counter number. All of the counters can be cleared using the hexadecimal sub-function code 0x0A.

The following sub-function codes are supported:

Sub Function	Name	Description
0x0A	Clear all counters	Sets all counters to zero
0x0B	Return Bus Message Count	No. of valid messages received (with all addresses)
0x0C	Return Bus Communication Error Count	No. of messages with CRC or Parity/Framing/Overrun error
0x0D	Return Bus Exception Error Count	No. of Exception Responses sent
0x0E	Return Slave Message Count	No. of messages received (with slave address)
0x0F	Return Slave No Response Count	No. of Broadcast messages received
0x10	Return Slave NAK Count	not used, return value is always 0
0x11	Return Slave Busy Count	not used, return value is always 0
0x12	Return Bus Character Overrun Count	No. of messages with Overrun error

Request (Sub-function 0x0A, Clear all Counters):

Start Frame (RTU oder ASCII mode)		
Address		
Function Code	1 Byte	0x08
Sub-function	2 Bytes	0x000A
Data	2 Bytes	0x0000
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address		
Function Code	1 Byte	0x08
Sub-function	2 Bytes	0x000A
Data	2 Bytes	0x0000
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address		

Error Code	1 Byte	0x88
Exception Code	1 Byte	1, 3 or 4
CRC or LRC & Endframe		

Data

This field is always 0x0000.

Exception Code

1 ILLEGAL FUNCTION CODE – Sub-function not supported

3 ILLEGAL DATA VALUE – No. of bytes in the data field too small or too large

– "data field" not equal to 0x0000

4 SLAVE DEVICE FAILURE – Error on carrying out the function

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

Request (Sub-function 0x0B – 0x12, Return Counter Value):

Start Frame (RTU oder ASCII mode)		
Address		
Function Code	1 Byte	0x08
Sub-function	2 Bytes	0x000B – 0x0012
Data	2 Bytes	0x0000
CRC or LRC & Endframe		

Response:

Start Frame (RTU oder ASCII mode)		
Address		
Function Code	1 Byte	0x08
Sub-function	2 Bytes	0x000B – 0x0012
Data	2 Bytes	0 – 0xFFFF
CRC or LRC & Endframe		

Exception Response:

Start Frame (RTU oder ASCII mode)		
Address		
Error Code	1 Byte	0x88
Exception Code	1 Byte	1, 3 or 4
CRC or LRC & Endframe		

Data

This field is always set to 0x0000 in the request and contains the actual counter value in the response.

Exception Code

The following exception codes can occur:

1 ILLEGAL FUNCTION CODE – Sub-function not supported

3 ILLEGAL DATA VALUE – No. of bytes in the data field too small or too large

– "data field" not equal to 0x0000

4 SLAVE DEVICE FAILURE – Error on Reading the Diagnostic Counter

Exception Codes are described in detail in Chapter 7.3.9 "Exception Codes".

6.3.8 Exception Responses

When the master device sends a request to the inverter it expects a normal response. One of four possible events can occur from the master’s query:

- If the inverter receives the request without a communication error and can handle the query normally, it returns a normal response.
- If the inverter does not receive the request due to a communication error, no response is returned. The master will eventually process a timeout condition for the request.
- If the inverter receives the request, but detects a communication error (parity, LRC, CRC, ...), no response is returned. The master will eventually process a timeout condition for the request.
- If the inverter receives the request without a communication error, but cannot handle it (for example, if the request is to read an unknown parameter), the inverter will return an exception response informing the client of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field:

In a normal response, the inverter echoes the function code of the original request in the function code field of the response. All function codes have a most–significant bit (MSB) of 0 (their values are all below 0x80 hexadecimal). In an exception response, the inverter sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 0x80 hexadecimal higher than the value would be for a normal response. With the function code’s MSB set, the master can recognize the exception response and can examine the data field for the exception code.

Data Field:

In a normal response, the inverter may return data or statistics in the data field (any information that was requested in the request). In an exception response, the inverter returns an exception code in the data field. This defines the server condition that caused the exception.

The Exception Codes generated by the inverter are listed in Chapter 7.3.9 "Exception Codes".

6.3.9 Exception Codes

The following exception codes are generated by the inverter:

Code	Modbus Name	Cases when generated by the Inverter
1	ILLEGAL FUNCTION	Function Code unknown Subfunction Code unknown (Diagnostics Function)
2	ILLEGAL DATA ADDRESS	No. of Registers field incorrect (must always be 0x01) Unknown Parameter or Parameter Data Type mismatch
3	ILLEGAL DATA VALUE	Error in dataset of frame No. of bytes too small or too large Certain fields not set to specific values
4	SLAVE DEVICE FAILURE	Read or Write Parameter failed The reason for the error can be obtained by reading out Parameter *VABusSST Error Register* **11**.

VABusSST Error Register **11**	
Error number	**Meaning**
0	no error
1	inadmissible parameter value
2	inadmissible data set
3	parameter not readable (write-only)
4	parameter not writable (read-only)
5	read error EEPROM
6	write error EEPROM
7	checksum error EEPROM
8	parameter cannot be written while the drive is running
9	values of the data sets differ from one another
10	wrong parameter type
11	unknown parameter
12	checksum error in received message
13	syntax error in received message
14	data type of parameter does not correspond to the number of bytes in the message
15	unknown error

When the error register *VABus SST Error Register* **11** is read out, it is automatically cleared at the same time.

6.3.10 Modbus Transmission Modes

Two different serial transmission modes are defined: the **RTU mode** and the **ASCII mode**. They define the bit contents of message fields transmitted serially on the line. They determine how information is packed into the message fields and decoded.

The transmission mode (and serial port parameters) must be the same for all devices on a MODBUS Serial Line.

The transmission mode for the Modbus-Communication can be selected via Parameter *Protocol* **395**. See Chapter 5.2.4 "Setting the Protocol".

6.3.10.1 RTU Transmission

When devices communicate on a MODBUS serial line using the RTU (Remote Terminal Unit) mode, each 8–bit byte in a message contains two 4–bit hexadecimal characters.

The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.

6.3.10.1.1 Character Format

One character consists of 11 bits.

- 1 start bit
- 8 data bits, least significant bit sent first
- 1 parity bit
- 1 stop bit

Note:
If no parity is used then an extra stop bit is added.

Character format with parity checking:

Start	B1	B2	B3	B4	B5	B6	B7	B8	Parity	Stop

Character format without parity checking:

Start	B1	B2	B3	B4	B5	B6	B7	B8	Stop	Stop

6.3.10.1.2 Frame Description

Slave Address	Function Code	Data	CRC
1 Byte	1 Byte	0 to 252 Bytes	2 Bytes (CRC-Lo, CRC-Hi)

The maximum size of a MODBUS RTU frame is 256 bytes.

6.3.10.1.3 Modbus Message RTU Framing

A MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. Partial messages must be detected and errors must be set as a result. In RTU mode, message frames are separated by a silent interval of at least 3.5 character times.

MODBUS Message

Start		Address	Function	Data	CRC		End
>= 3,5 Char		8 Bits	8 Bits	N x 8 Bits	16 Bits		>= 3,5 Char

The entire message frame must be transmitted as a contiguous stream of characters.

If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and is discarded by the inverter.

CRC Checking

The RTU mode includes an error–checking field that is based on a Cyclical Redundancy Checking (**CRC**) method performed on the message contents.

The CRC field checks the contents of the entire message. It is applied regardless of any parity checking method used for the individual characters of the message.

The CRC field contains a 16–bit value implemented as two 8–bit bytes.

The CRC field is appended to the message as the last field in the message. It is appended in the sequence "low-order byte", "high-order byte". The CRC high–order byte is the last byte to be sent in the message.

The CRC value is calculated by the sending device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC calculation is started by first pre-loading a 16–bit register to all 1's (0xFFFF). Then a process begins of applying successive bytes of the message to the current contents of the register.

During generation of the CRC, each 8bit character is ExOR with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then ExOR with the fixed value *0xA001*. If the LSB was a 0, no ExOR takes place.

This process is repeated until eight shifts have been performed. After the last (eight) shift, the next 8bit char is ExOR with the register's current value, and the process repeats for eight more shifts as described above. The final content of the register, after all the bytes of the message have been applied, is the CRC value.

When the CRC is appended to the message, the low-order byte is appended first, followed by the high-order byte.

Example for CRC calculating for the frame: "0x02 0x07"

	MSB			**LSB**	
Start CRC	1111	1111	1111	1111	0xFFFF
1.char	0000	0000	0000	0010	**0x02**
ExOR					
CRC	1111	1111	1111	1101	
1.shift right	0111	1111	1111	1110	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1101	1111	1111	1111	
2.shift right	0110	1111	1111	1111	
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1100	1111	1111	1110	
3.shift right	0110	0111	1111	1111	⇒0 = no ExOR
4.shift right	0011	0011	1111	1111	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1001	0011	1111	1110	
5.shift right	0100	1001	1111	1111	⇒0 = no ExOR
6.shift right	0010	0100	1111	1111	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1000	0100	1111	1110	
7.shift right	0100	0010	0111	1111	⇒0 = no ExOR
8.shift right	0010	0001	0011	1111	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1000	0001	0011	1110	
2.char	0000	0000	0000	0111	**0x07**
ExOR					
CRC	1000	0001	0011	1001	
1.shift right	0100	0000	1001	1100	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1110	0000	1001	1101	

2. shift right	0111	0000	0100	1110	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1101	0000	0100	1111	
3. shift right	0110	1000	0010	0111	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1100	1000	0010	0110	
4. shift right	0110	0100	0001	0011	⇒0 = no ExOR
5. shift right	0011	0010	0000	1001	⇒1 = ExOR
0xA001	1010	0000	0000	0001	
ExOR					
CRC	1001	0010	0000	1000	
6.shift right	0100	1001	0000	0100	⇒0 = no ExOR
7.shift right	0010	0100	1000	0010	⇒0 = no ExOR
8.shift right	0001	0010	0100	0001	⇒0 = no ExOR
result	**1**	**2**	**4**	**1**	CRC = 0x41 0x12

Example: program sequence (C#)

```
Private int Modbus_CRC(string frame)
{
  int poly     = 0xA001;                              // polynom
  int CRC      = 0xFFFF;                              // start CRC

  for (int i = 0; i < (int)(frame.Length); i++)      // for each char
  {
    CRC ^= Convert.ToInt16(frame[i]);                 //  ExOR

    for (int j = 0; j < 8; j++)                       //  eight times
    {
      if ((CRC& 0x01) == 0x01)                        //  lsb == 1 ?
      {
       CRC>>= 1;                                      //  shift left
       CRC ^= poly;                                   //  ExOR
      }
      else
      {
        CRC>>= 1;                                     //  shift left
      }
    }
  }
  Return CRC;
}
```

Call the function for the frame "0x02 0x07"

```
stringtemp_s = Convert.ToString((char)(0x2))+ Convert.ToString((char)(0x7));

Console.WriteLine(Modbus_CRC(temp_s));
```

Result:	4673	= 0x1241
CRC Low Byte		= 0x41
CRC High Byte		= 0x12

The Modbus CRC sequence is:

⇒	CRC	Low Byte/High Byte	0x41	0x12

6.3.10.2 ASCII Transmission

When devices are setup to communicate on a MODBUS serial line using 7-bit ASCII (American Standard Code for Information Interchange) mode, each 8 bit character in a message is sent as two ASCII characters. This mode is used when the physical communication link or the capabilities of the device does not allow the conformance with RTU mode requirements regarding the management of timers.

Note:
This mode is less efficient than RTU since each byte needs two characters.

Example: The byte 0x5B is encoded as two characters: 0x35 and 0x42 (0x35 ="5", and 0x42 ="B" in ASCII).

6.3.10.2.1 Character Format

One character consists of 10 bits:

- 1 start bit
- 7 data bits, least significant bit sent first (value = 0x00 .. 0x7F)
- 1 parity bit
- 1 stop bit

Note:
If no parity is used then an extra stop bit is added.

Character format with parity checking:

Start	B1	B2	B3	B4	B5	B6	B7	Parity	Stop

Character format without parity checking:

Start	B1	B2	B3	B4	B5		B6	B7	Stop	Stop

6.3.10.2.2 Modbus Message ASCII Framing

A MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. Partial messages must be detected and errors must be set as a result.

The address field of a message frame contains two characters.

In ASCII mode, a message is delimited by specific characters as Start-of-Frame and End-of-Frame. A message must start with a trigger "colon"- character (":" = ASCII 0x3A) and ends with a "carriage return/line feed" pair (ASCII 0x0D and 0x0A).

The allowable characters transmitted for all other fields are hexadecimal 0–9, A–F (ASCII coded). The devices monitor the bus continuously for the colon character. When this character is received, each device decodes the next character until it detects the End-of-Frame (EoF)-character.

Intervals of up to one second may elapse between characters within the message. Unless the user has configured a longer timeout, an interval greater than 1 second means an error has occurred.

A typical message frame is shown below:

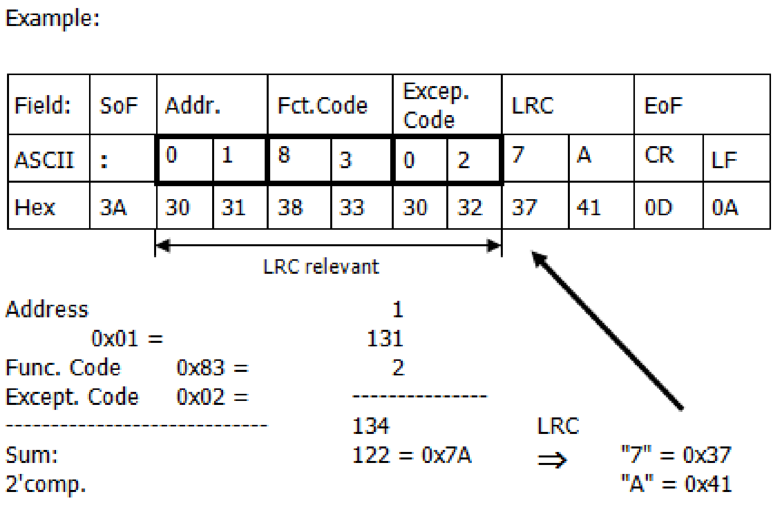
SoF	Address	Function	Data	LRC	EoF
1 char :	2 chars	2 chars	0 up to 2* 252 chars	2 chars	2 chars CR, LF

6.3.10.2.3 LRC Checking

In ASCII mode, messages include an error–checking field that is based on a Longitudinal Redundancy Checking (**LRC**) calculation that is performed on the message contents, exclusive of the beginning ‘colon’ and terminating CRLF pair characters. It is applied regardless of any parity checking method used for the individual characters of the message.

The LRC field is one byte, containing an 8-bit binary value. The LRC value is calculated by the device that emits, which appends the LRC to the message. The device that receives calculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results.

The LRC field is one byte, containing an 8-bit binary value. The LRC value is calculated by the device that emits, which appends the LRC to the message. The device that receives calculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results.



Example:

Field:	SoF	Addr.		Fct.Code		Excep. Code		LRC		EoF	
ASCII	:	0	1	8	3	0	2	7	A	CR	LF
Hex	3A	30	31	38	33	30	32	37	41	0D	0A

LRC relevant

```
Address                              1
          0x01 =                   131
Func. Code       0x83 =              2
Except. Code     0x02 =         ---------------
-------------------------       134             LRC
Sum:                            122 = 0x7A      ⇒      "7" = 0x37
2'comp.                                                "A" = 0x41
```

6.4 Time Monitoring Function

The Modbus protocol defines a pure Master/Slave operation. If a frequency inverter is addressed by the bus master, other frequency inverters will only be addressed after the protocol with the first frequency inverter has been completed or the timeout time has expired.

After a frequency inverter has sent a message, a waiting time of **2 ms** must be kept. This time is needed by the frequency inverter in order to switch off the RS485 transmitter. The bus master may not send a new message until this time has elapsed.

Attention!
If the degree of utilization of the CPU is high (> 90%), the response time can be more than 500 ms.

Note:
The specified times are valid for operation with RS485 and RS232.

7 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 the values of which exist once (data set 0) as well as parameters the values of which exist four times (data set 1...4). These are used for data set change-over.

If parameters which exist four times in the data sets are set to data set = 0, all four data sets are set to the same 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, error 9 = "different values in data sets" is signaled via the error register *VABus SST Error Register* **11**. In this case, you must read out each data set separately for the relevant parameter (see chapter 7.3.9 "Exception Codes").

New write requests will be blocked by the error register *VABus SST Error Register* **11**. For this reason, the error register must be read out, i.e. acknowledged, before a new write request can be sent.

Regardless of the signal status of the error register, reading access (read requests) is still possible.

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

Caution!

If the number of admissible writing cycles is exceeded, the EEPROM is destroyed.

In order to avoid the destruction of the EEPROM, data which are to be written cyclically can be entered in the RAM exclusively without a writing cycle on the EEPROM. In this case, the data are volatile, i.e. they are lost when the supply voltage is switched off (Mains Off). They must be written into the RAM again after the restart (Mains On).

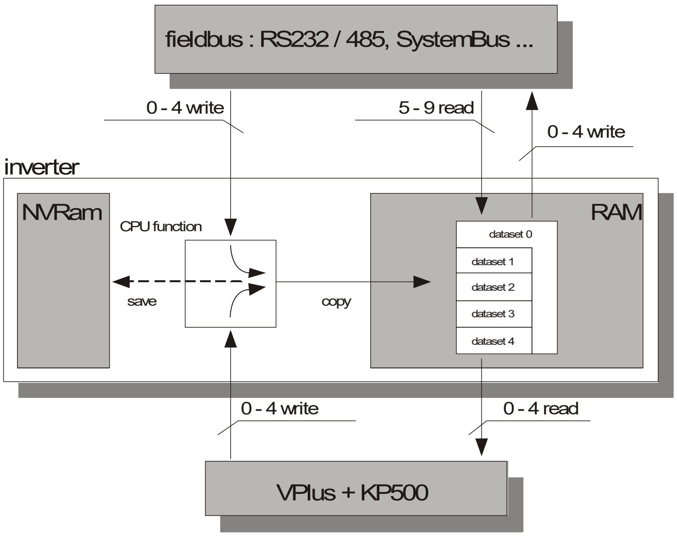
The RAM writing operation is activated by increasing the number of the target data set by five.

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

Note:

The Data Sets for parameters *Control Word* **410**, *Reference Frequency RAM* **484** and *Reference Percentage RAM* **524** are always zero. Internally, these are not written to the EEPROM.


fieldbus : RS232 / 485, SystemBus ...
0 - 4 write
5 - 9 read
0 - 4 write
inverter
NVRam
CPU function
save
copy
RAM
dataset 0
dataset 1
dataset 2
dataset 3
dataset 4
0 - 4 write
0 - 4 read
VPlus + KP500

8 Modbus Message Examples

This chapter describes some examples for Modbus RTU and Modbus ASCII.

8.1 Modbus RTU-Message Examples

8.1.1 16 Bit Parameter access

8.1.1.1 Function Code 3, Read 16-Bit Parameter

Example 1:

Reading parameter *Rated Speed* **372** (0x0174) in data set 2 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		No. of Regs		CRC	
Hex	01	03	21	74	00	01	CE	2C

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	No.Bytes	Par-Value		CRC	
Hex	01	03	02	05	6E	3A	F8

The hexadecimal value sent is 0x056E = 1390. Parameter *Rated Speed* **372** has no decimal places.

Therefore, the rated speed is 1390 min^{-1}.

Example 2:

Reading parameter *Rated Speed* **372** (0x0174) in data set 0 from the frequency inverter with address 1 and No. of Regs field set to 2 (invalid value).

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		No. of Regs		CRC	
Hex	01	03	01	74	00	02	85	ED

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	01	83	04	40	F3

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.1.2 Function Code 6, Write 16-Bit Parameter

Example 1:

Writing parameter *Rated Mech. Power* **376** (0x0178) in data set 4 to the frequency inverter with address 3.

The rated mechanical power is to be set to 1.5 kW. Parameter *Rated Mech. Power* **376** has one decimal place.

Therefore the value to be sent is 15 = 0x000F.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Funk.	DSatz/Par-Nr.		Par-Wert		CRC	
Hex	03	06	41	78	00	0F	5C	09

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	DSet/Par-No.		Par-Value		CRC	
Hex	03	06	41	78	00	0F	5C	09

The response is an echo of the request message.

Example 2:

Writing out of range value 0 to parameter **376** (0x0178) in data set 2 to the frequency inverter with address 3.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		Par-Value		CRC	
Hex	03	06	21	78	00	00	02	0D

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	03	86	04	E2	63

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.1.3 Function Code 16, Write 16-Bit Parameter

Example 1:

Writing parameter *Rated Mech. Power* **376** (0x0178) in data set 4 to the frequency inverter with address 1.

The rated mechanical power is to be set to 1.5 kW. Parameter *Rated Mech. Power* **376** has one decimal place.

Therefore the value to be sent is 15 = 0x000F.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSatz/Par-Nr.		No. of registers		No. of bytes	Par-value		CRC	
Hex	01	10	41	78	00	01	02	00	0F	BD	28

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	DSet/Par-No.		No. of registers		CRC	
Hex	01	10	41	78	00	01	95	EC

The response contains the no. of registers written.

Example 2:

Writing out of range value 0 to parameter *Rated Mech. Power* **376** (0x0178) in data set 2 to the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSatz/Par-Nr.		No. of registers		No. of bytes	Par-value		CRC	
Hex	03	10	41	78	00	01	02	00	00	E4	4C

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	03	90	04	EC	03

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.2 32 Bit Zugriff

8.1.2.1 Function Code 3, Read 32-Bit Parameter

Example 1:

Reading parameter *Fixed Frequency 2* **481** (0x01E1) in data set 1 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		No. of Regs		CRC	
Hex	01	03	11	E1	00	02	91	01

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	No.Bytes	Par-Value				CRC	
Hex	01	03	04	00	00	03	E8	FA	8D

The hexadecimal value sent is 0x03E8 = 1000. Parameter *Fixed Frequency* **481** has two decimal places.

Therefore, the frequency is 10.00 Hz.

Example 2:

Reading parameter *Fixed Frequency 2* **481** (0x01E1) in data set 0 from the frequency inverter with address 1 and No. of Regs field set to 1 (invalid value).

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		No. of Regs		CRC	
Hex	01	03	01	E1	00	01	D5	C0

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	01	83	04	40	F3

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.2.2 Function Code 16, Write 32-Bit Parameter

Example 1:

Writing parameter *Fixed Frequency 3* **482** (0x01E2) in data set 9 (= RAM for data set4) to the frequency inverter with address 1.

The frequency is to be set to 44.50 Hz. Parameter *Fixed Frequency 3* **482** has two decimal places.

Therefore the value to be sent is 4450 = 0x00001162.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Funk.	DSatz/Par-Nr.		No. Registers		No. Bytes	Par-Wert				CRC	
Hex	01	10	91	E2	00	02	04	00	00	11	62	50	11

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	DSet/Par-No.		No. Registers		CRC	
Hex	01	10	91	E2	00	02	CD	02

The response contains the no. of registers written.

Example 2:

Writing parameter *Fixed Frequency 3* **482** (0x01E2) in data set 9 (= RAM for data set4) to the frequency inverter with address 1.

The frequency is to be set to the invalid value 2000.00 Hz. Parameter *Fixed Frequency 3* **482** has two decimal places.

Therefore the value to be sent is 20000 = 0x00030D40.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Funk.	DSatz/Par-Nr.		No. Registers		No. Bytes	Par-Wert				CRC	
Hex	01	10	91	E2	00	02	04	00	03	0D	40	28	C8

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	03	90	04	4D	C3

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.2.3 Function Code 100, Read 32-Bit Parameter

Example 1:

Reading parameter *Fixed Frequency 2* **481** in data set 0 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Funk.	DSatz/Par-Nr.		CRC	
Hex	01	64	01	E1	81	DF

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Par-Value				CRC	
Hex	01	64	00	00	03	E8	70	BC

The hexadecimal value sent is 0x000003E8 = 1000. Parameter *Fixed Frequency 2* **481** has two decimal places.

Therefore, fixed frequency 2 = 10.00 Hz.

Example 2:

Reading unknown parameter **1600** (0x0640) in data set 2 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		CRC	
Hex	01	64	26	40	5B	97

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	01	E4	04	6A	C3

Reading unknown parameter **1600** (0x0640) in data set 2 from the frequency inverter with address 1.

8.1.2.4 Function Code 101, Write 32-Bit Parameter

Example 1:

Writing parameter *Rated Frequency* **375** (0x0177) in data set 2 to the frequency inverter with address 1.

The rated frequency is to be set to 10.00Hz. Parameter *Rated Frequency* **375** has two decimal places.

Therefore the value to be sent is 1000 = 0x03E8.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		Par-Value				CRC	
Hex	01	65	21	77	00	00	03	E8	46	C5

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	DSet/Par-No.		Par-Value				CRC	
Hex	01	65	21	77	00	00	03	E8	46	C5

The response is an echo of the request message.

Example 2:

Writing out of range value 9.00 Hz to parameter **375** in data set 2 of the frequency inverter with address 1.

Parameter *Rated Frequency* **375** has 2 decimal places. The value to be sent is 900 = 0x0384.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	DSet/Par-No.		Par-Value				CRC	
Hex	01	65	21	77	00	00	03	84	46	E8

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	01	E5	04	6B	53

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.1.3 Function Code 8, Diagnostics

Example 1a:

Clearing all Diagnostic Counters (Sub-function 0x0A) in the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	Sub-function		Data		CRC	
Hex	01	08	00	0A	00	00	C0	09

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Sub-function		Data		CRC	
Hex	01	08	00	0A	00	00	C0	09

The response is an echo of the request. All counters are now set to zero.

Example 1b:

With all counters set to zero, read Diagnostic Counter 4, "Slave Message Count" (Sub-function 0x0E) from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Func.	Sub-function		Data		CRC	
Hex	01	08	00	0E	00	00	81	C8

Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Sub-function		Data		CRC	
Hex	01	08	00	0E	00	01	40	08

The counter value is 1(as this is the 1st message received after setting all counters to zero).

Example 2:

Reading unknown Diagnostic Counter 8 (Sub-function 0x13) from frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	Addr.	Funk.	Sub-function		Data		CRC	
Hex	01	08	00	13	00	00	11	CE

Error Response: Frequency Inverter ➔ Master

Field:	Addr.	Func.	Except.	CRC	
Hex	01	88	01	87	C0

The exception code sent is hexadecimal value 0x01 = ILLEGAL FUNCTION.

8.2 Modbus ASCII-Message Examples

8.2.1 16 Bit access

8.2.1.1 Function Code 3, Read 16-Bit Parameter

Example 1:

Reading parameter *Rated Speed* **372** (0x0174) in data set 2 from the frequency inverter with address 1.

Request: Master → Frequency Inverter

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. of Regs				LRC		EoF	
ASC	:	0	1	0	3	2	1	7	4	0	0	0	1	6	6	CR	LF
Hex	3A	30	31	30	33	32	31	37	34	30	30	30	31	36	36	0D	0A

Response: Frequency Inverter → Master

Field:	SoF	Addr.		Fct.Code		No. of Bytes		Par-Wert				LRC		EoF	
ASC	:	0	1	0	3	0	2	0	5	6	E	8	7	CR	LF
Hex	3A	30	31	30	33	30	32	30	35	36	45	38	37	0D	0A

The hexadecimal value sent is 0x056E = 1390. Parameter *Rated Speed* **372** has no decimal places.

Therefore, the rated speed is 1390 min^{-1}.

Example 2:

Reading parameter *Rated Speed* **372** (0x0174) in data set 0 from the frequency inverter with address 1 and No. of Regs field set to 2 (invalid value).

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. of Regs				LRC		EoF	
ASC	:	0	1	0	3	0	1	7	4	0	0	0	2	8	5	CR	LF
Hex	3A	30	31	30	33	30	31	37	34	30	30	30	32	38	35	0D	0A

Error Response: Frequency Inverter → Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	8	3	0	4	7	8	CR	LF
Hex	3A	30	31	38	33	30	34	37	38	0D	0A

The sent exception code is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.1.2 Function Code 6, Write 16-Bit Parameter

Example 1:

Writing parameter *Rated Mech. Power* **376** (0x0178) in data set 4 to the frequency inverter with address 3.

The rated mechanical power is to be set to 1.5 kW. Parameter *Rated Mech. Power* **376** has one decimal place.

Thus the value to be sent is 15 = 0x000F.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				Para-Value				LRC		EoF	
ASC	:	0	3	0	6	4	1	7	8	0	0	0	F	2	F	CR	LF
Hex	3A	30	33	30	36	34	31	37	38	30	30	30	46	32	46	0D	0A

Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				Para-Value				LRC		EoF	
ASC	:	0	3	0	6	4	1	7	8	0	0	0	F	2	F	CR	LF
Hex	3A	30	33	30	36	34	31	37	38	30	30	30	46	32	46	0D	0A

The response is an echo of the request message.

Example 2:

Writing out of range value 0 to parameter *Rated Mech. Power* **376** (0x0178) in data set 2 of the frequency inverter with address 3.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				Para-Value				LRC		EoF	
ASC	:	0	3	0	6	2	1	7	8	0	0	0	0	5	E	CR	LF
Hex	3A	30	33	30	36	32	31	37	38	30	30	30	30	35	45	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	3	8	6	0	4	7	3	CR	LF
Hex	3A	30	33	38	36	30	34	37	33	0D	0A

The sent exception code is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.1.3 Function Code 16, Write 16-Bit Parameter

Example 1:

Writing parameter *Rated Mech. Power* **376** (0x0178) in data set 4 to the frequency inverter with address 3.

The rated mechanical power is to be set to 1.5 kW. Parameter *Rated Mech. Power* **376** has one decimal place.

Thus the value to be sent is 15 = 0x000F.

Request: Master ➔ Frequency Inverter

Field	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				No. Byte		Para-Value				LRC		EoF	
ASC	:	0	1	1	0	4	1	7	8	0	0	0	1	0	2	0	0	0	F	2	4	CR	LF
Hex	3A	30	31	31	30	34	31	37	38	30	30	30	31	30	32	30	30	30	46	32	34	0D	0A

Response: Frequency Inverter ➔ Master

Field	SoF	Addr.		Fct.Code		DSet/Par-No.				Para-Value				LRC		EoF	
ASC	:	0	1	1	0	4	1	7	8	0	0	0	F	3	5	CR	LF
Hex	3A	30	31	31	30	34	31	37	38	30	30	30	46	33	35	0D	0A

Example 2:

Writing out of range value 0 to parameter *Rated Mech. Power* **376** (0x0178) in data set 2 of the frequency inverter with address 3.

Request: Master ➔ Frequency Inverter

Field	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				No. Byte		Para-Value				LRC		EoF	
ASC	:	0	1	1	0	2	1	7	8	0	0	0	1	0	2	0	0	0	0	5	3	CR	LF
Hex	3A	30	33	30	36	32	31	37	38	30	30	30	31	30	32	30	30	30	30	35	33	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	9	0	0	4	6	B	CR	LF
Hex	3A	30	31	39	30	30	34	36	42	0D	0A

The sent exception code is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.2 32 Bit access

8.2.2.1 Function Code 3, Read 32-Bit Parameter

Example 1:

Reading parameter *Fixed Frequency 2* **481** (0x01E1) in data set 1 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Feld:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				LRC		EoF	
ASC	:	0	1	0	3	1	1	E	1	0	0	0	2	0	8	CR	LF
Hex	3A	30	31	30	33	31	31	45	31	30	30	30	32	30	38	0D	0A

Response: Frequency Inverter ➔ Master

Feld:	SoF	Addr.		Fct.Code		No. Byte		Para-Value								LRC		EoF	
ASC	:	0	1	0	3	0	4	0	0	0	0	0	3	E	8	0	3	CR	LF
Hex	3A	30	31	31	30	30	34	30	30	30	30	30	33	45	38	30	33	0D	0A

The hexadecimal value sent is 0x03E8 = 1000. Parameter *Fixed Frequency* **481** has two decimal places.

Therefore, the frequency is 10.00 Hz.

Example 2:

Reading parameter *Fixed Frequency 2* **481** (0x01E1) in data set 0 from the frequency inverter with address 1 and No. of Regs field set to 1 (invalid value).

Request: Master ➔ Frequency Inverter

Feld:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				LRC		EoF	
ASC	:	0	1	0	3	0	1	E	0	0	0	0	1	1	A	CR	LF
Hex	3A	30	31	30	33	30	31	45	30	30	30	30	31	31	41	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	8	3	0	4	7	8	CR	LF
Hex	3A	30	31	38	33	30	34	37	38	0D	0A

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.2.2 Function Code 16, Write 32-Bit Parameter

Example 1:

Reading parameter *Fixed Frequency 3* **482** (0x01E2) in data set 0 from the frequency inverter with address 1.

The rated mechanical power is to be set to 44.50 Hz. Parameter *Fixed Frequency 3* **482** has two decimal places.

Thus the value to be sent is 4450 = 0x00001162.

Request: Master ➔ Frequency Inverter

Feld:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				No. Byte		Para-Value								LRC		EoF	
ASC	:	0	1	1	0	9	1	E	2	0	0	0	2	0	4	0	0	0	0	1	1	6	2	0	3	CR	LF
Hex	3A	30	31	31	30	39	31	45	32	30	30	30	32	30	34	30	30	30	30	31	31	36	32	30	33	0D	0A

Response: Frequency Inverter ➔ Master

Feld:	SoF	Addr.		Fct.Code		DSet/Par-No.				No. Registers				LRC		EoF	
ASC	:	0	1	1	0	9	1	E	2	0	0	0	2	7	A	CR	LF
Hex	3A	30	31	31	30	39	31	45	32	30	30	30	32	37	41	0D	0A

Example 2:

Writing parameter *Fixed Frequency 3* **482** (0x01E2) in data set 9 (= RAM for data set4) to the frequency inverter with address 1.

The frequency is to be set to the invalid value 2000.00 Hz. Parameter *Fixed Frequency 3* **482** has two decimal places.

Therefore the value to be sent is 20000 = 0x00030D40.

Request: Master ➔ Frequency Inverter

Field	SoF	Addr.		Fct.Co de		DSet/Par-No.				No. Registers				No. Byte		Para-Value				LRC		EoF	
ASC	:	0	1	1	0	4	1	7	8	0	0	0	1	0	2	0	0	0	0	3	3	CR	LF
Hex	3A	30	33	30	36	32	31	37	38	30	30	30	31	30	32	30	30	30	30	33	33	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	9	0	0	4	6	B	CR	LF
Hex	3A	30	31	39	30	30	34	36	42	0D	0A

The sent exception code is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.2.3 Function Code 100, Read 32-Bit Parameter

Example 1:

Reading parameter *Fixed Frequency 2* **481** in data set 0 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				LRC		EoF	
ASC	:	0	1	6	4	0	1	E	1	B	9	CR	LF
Hex	3A	30	31	36	34	30	31	45	31	42	39	0D	0A

Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Para-Value								LRC		EoF	
ASC	:	0	1	6	4	0	0	0	0	0	3	E	8	B	0	CR	LF
Hex	3A	30	31	36	34	30	30	30	30	30	33	45	38	42	30	0D	0A

The hexadecimal value sent is 0x000003E8 = 1000. Parameter *Fixed Frequency 2* **481** has 2 decimal places.

Therefore, fixed frequency 2 = 10.00 Hz.

Example 2:

Reading unknown parameter **1600** (0x0640) in data set 2 from the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		DSet/Par-No.				LRC		EoF	
ASC	:	0	1	6	4	2	6	4	0	3	5	CR	LF
Hex	3A	30	31	36	34	32	36	34	30	33	35	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	E	4	0	2	1	9	CR	LF
Hex	3A	30	31	45	34	30	34	31	37	0D	0A

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.2.4 Function Code 101, Write 32-Bit Parameter

Example 1:

Writing parameter *Rated Frequency* **375** (0x0177) in data set 2 to frequency inverter with address 1.

The rated frequency is to be set to 10.00 Hz. Parameter *Rated Frequency* **375** has two decimal places.

Therefore the value to be sent is 1000 = 0x03E8.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Cod		DSet/Par-No.				Para-Value								LRC		EoF	
ASC	:	0	1	6	5	2	1	7	7	0	0	0	0	0	3	E	8	1	7	CR	LF
Hex	3A	30	31	36	35	32	31	37	37	30	30	30	30	30	33	45	38	31	37	0D	0A

Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Cod		DSet/Par-No.				Para-Value								LRC		EoF	
ASC	:	0	1	6	5	2	1	7	7	0	0	0	0	0	3	E	8	1	7	CR	LF
Hex	3A	30	31	36	35	32	31	37	37	30	30	30	30	30	33	45	38	31	37	0D	0A

The response is an echo of the request message.

Example 2:

Writing out of range value to 9.00 Hz to parameter *Rated Frequency* **375** (0x0177) in data set 2 of the frequency inverter with address 1.

Parameter *Rated Frequency* **375** has 2 decimal places. The value to be sent is 900 = 0x0384

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Cod		DSet/Par-No.				Para-Value								LRC		EoF	
ASC	:	0	1	6	5	2	1	7	7	0	0	0	0	0	3	8	4	7	B	CR	LF
Hex	3A	30	31	36	35	32	31	37	37	30	30	30	30	30	33	38	34	37	42	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	E	5	0	4	1	6	CR	LF
Hex	3A	30	31	45	35	30	34	31	36	0D	0A

The exception code sent is hexadecimal value 0x04 = SLAVE DEVICE FAILURE.

8.2.3 Function Code 8, Diagnostics

Example 1a:

Clearing all Diagnostic Counters (Sub-function 0x0A) in the frequency inverter with address 1.

Request: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		Sub-function				Data				LRC		EoF	
ASC	:	0	1	0	8	0	0	0	A	0	0	0	0	E	D	CR	LF
Hex	3A	30	31	30	38	30	30	30	41	30	30	30	30	45	44	0D	0A

Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		DSet/ Para-No.				No. of Regs				LRC		EoF	
ASC	:	0	1	0	8	0	0	0	A	0	0	0	0	E	D	CR	LF
Hex	3A	30	31	30	38	30	30	30	41	30	30	30	30	45	44	0D	0A

The response is an echo of the request. All counters are now set to zero.

Example 1b:

With all counters set to zero, read Diagnostic Counter 4, "Slave Message Count" (Sub-function 0x0E) from the frequency inverter with address 1.

Field:	SoF	Addr.		Fct.Code		Sub-function				Data				LRC		EoF	
ASC	:	0	1	0	8	0	0	0	E	0	0	0	0	E	9	CR	LF
Hex	3A	30	31	30	38	30	30	30	45	30	30	30	30	45	39	0D	0A

Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Sub-function				Data				LRC		EoF	
ASC	:	0	1	0	8	0	0	0	E	0	0	0	1	E	8	CR	LF
Hex	3A	30	31	30	38	30	30	30	45	30	30	30	31	45	38	0D	0A

Example 2:

Read unknown Diagnostic Counter 8 (Sub-function 0x13) from frequency inverter with address 1.

Response: Master ➔ Frequency Inverter

Field:	SoF	Addr.		Fct.Code		Sub-function				Data				LRC		EoF	
ASC	:	0	1	0	8	0	0	1	3	0	0	0	0	E	4	CR	LF
Hex	3A	30	31	30	38	30	30	31	33	30	30	30	30	45	34	0D	0A

Error Response: Frequency Inverter ➔ Master

Field:	SoF	Addr.		Fct.Code		Except. Code		LRC		EoF	
ASC	:	0	1	8	8	0	1	7	6	CR	LF
Hex	3A	30	31	38	38	30	31	37	36	0D	0A

The exception code sent is hexadecimal value 0x01 = ILLEGAL FUNCTION.

8.3 ASCII-Table (0x00 – 0x7F)

Dec.	Hex.	Char.	Dec.	Hex.	Char.	Dec.	Hex.	Char.
0	00	NUL	43	2B	+	86	56	V
1	01		44	2C	,	87	57	W
2	02	STX	45	2D	-	88	58	X
3	03	ETX	46	2E	.	89	59	Y
4	04	EOT	47	2F	/	90	5A	Z
5	05	ENQ	48	30	0	91	5B	[
6	06	ACK	49	31	1	92	5C	\
7	07	BEL	50	32	2	93	5D	]
8	08	BS	51	33	3	94	5E	^
9	09	TAB	52	34	4	95	5F	_
10	0A	LF	53	35	5	96	60	`
11	0B	VT	54	36	6	97	61	a
12	0C	FF	55	37	7	98	62	b
13	0D	CR	56	38	8	99	63	c
14	0E		57	39	9	100	64	d
15	0F		58	3A	:	101	65	e
16	10		59	3B	;	102	66	f
17	11		60	3C	<	103	67	g
18	12		61	3D	=	104	68	h
19	13		62	3E	>	105	69	i
20	14		63	3F	?	106	6A	j
21	15	NAK	64	40	@	107	6B	k
22	16		65	41	A	108	6C	l
23	17		66	42	B	109	6D	m
24	18		67	43	C	110	6E	n
25	19		68	44	D	111	6F	o
26	1A		69	45	E	112	70	p
27	1B	ESC	70	46	F	113	71	q
28	1C		71	47	G	114	72	r
29	1D		72	48	H	115	73	s
20	1E		73	49	I	116	74	t
31	1F		74	4A	J	117	75	u
32	20	SPACE	75	4B	K	118	76	v
33	21	!	76	4C	L	119	77	w
34	22	„	77	4D	M	120	78	x
35	23	#	78	4E	N	121	79	y
36	24	$	79	4F	O	122	7A	z
37	25	%	80	50	P	123	7B	{
38	26	&	81	51	Q	124	7C	|
39	27	'	82	52	R	125	7D	}
40	28	(	83	53	S	126	7E	~
41	29	)	84	54	T	127	7F	DEL
42	2A	*	85	55	U			

Values used often are highlighted in grey.

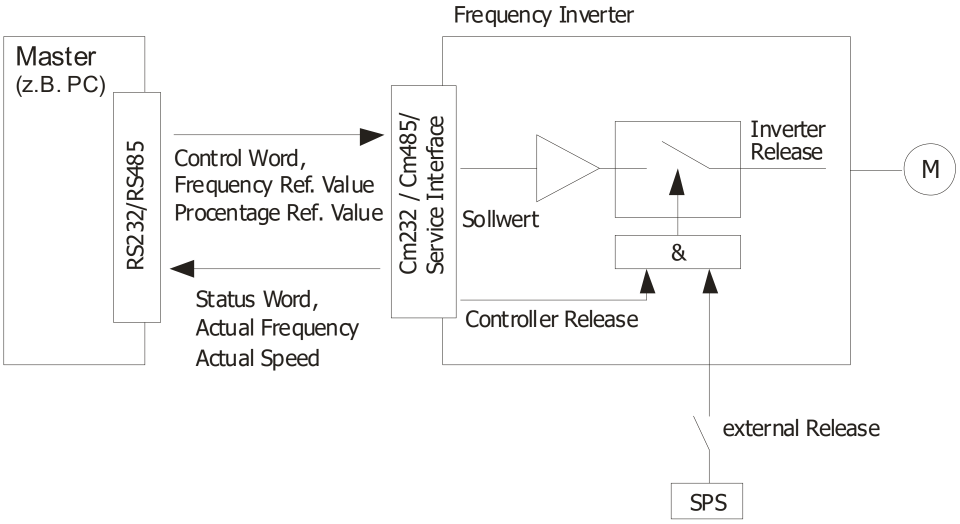
9 Control / Reference Value

- *410 Control Word*
- *411 Status Word*
- *484 Reference Frequency RAM [Hz]*
- *524 Reference Percentage RAM [%]*

The frequency inverter can be controlled completely via the serial interface. The following parameters and actual values are used for this:

Parameter		Setting			
No.	**Name/Meaning**	**Min.**	**Max.**	**Factory Setting**	**Type**
410	Control Word	0x0000	0xFFFF	-	uInt
411	Status Word	0x0000	0xFFFF	-	uInt
484	Ref. Frequency Value RAM [Hz]	-999,99	999,99	0,00	Long
524	Ref. Percentage RAM [%]	-300,00	300,00	0,00	Long

With the *Control Word* **410** (data type uInt), control commands are sent to the frequency inverter. With the *Reference Frequency RAM* **484** (data type Long [Hz]) or *Reference Percentage RAM* **524** (data type Long [%]), the reference line value is sent. Via the *Status Word* **411** (data type uInt), the status of the frequency inverter is read out.



Note:

Control Word **410**, *Reference Frequency RAM* **484** and *Reference Percentage RAM* **524** are stored in the RAM of the frequency inverter. This is generally addressed via data set 0.

- *412 Local/Remote*

The frequency inverter can be controlled with various operation modes. These operation modes can be selected with parameter *Local/Remote* **412**.

Local/Remote **412**		Function
0 -	Control via Contacts	The Start and Stop commands as well as the direction of rotation are set via digital signals.
1 -	Control via Statemachine	The Start and Stop commands as well as the direction of rotation are controlled via the Remote Statemachine[1] of the communication interface.
2 -	Control via Remote Contacts	The Start and Stop commands as well as the direction of rotation are controlled via virtual digital signals through the communication protocol.

For operation via the serial interface, settings 0, 1 and 2 are relevant. Further possible operation modes *Local/Remote* **412** are described in the frequency inverter operating instructions. These relate to the control via the Operator Panel and the control via digital signals.

[1] Statemachine is a standardized software module within the controller of the frequency inverter. The Statemachine represents specified operating states and control within the frequency inverter.

Parameter *Local/Remote* **412** is data set related, i.e. by selecting a data set, you can switch over between the different operation modes.

- *414 Data Set Selection*

The data set switch-over can be carried out via control contacts at the digital inputs of the frequency inverter or via the bus. For data set change-over via the bus, parameter *Data Set Selection* **414** is used.

Parameter		Setting		
No.	**Name**	**Min.**	**Max.**	**Factory Setting**
414	Data Set Selection	0	5	0

With the default setting *Data Set Selection* **414** = 0, the data set change-over is carried out via the digital inputs.

If *Data Set Selection* **414** is set to 1, 2, 3 or 4, the selected data set is activated via the bus. At the same time, data set change-over via the digital 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**.

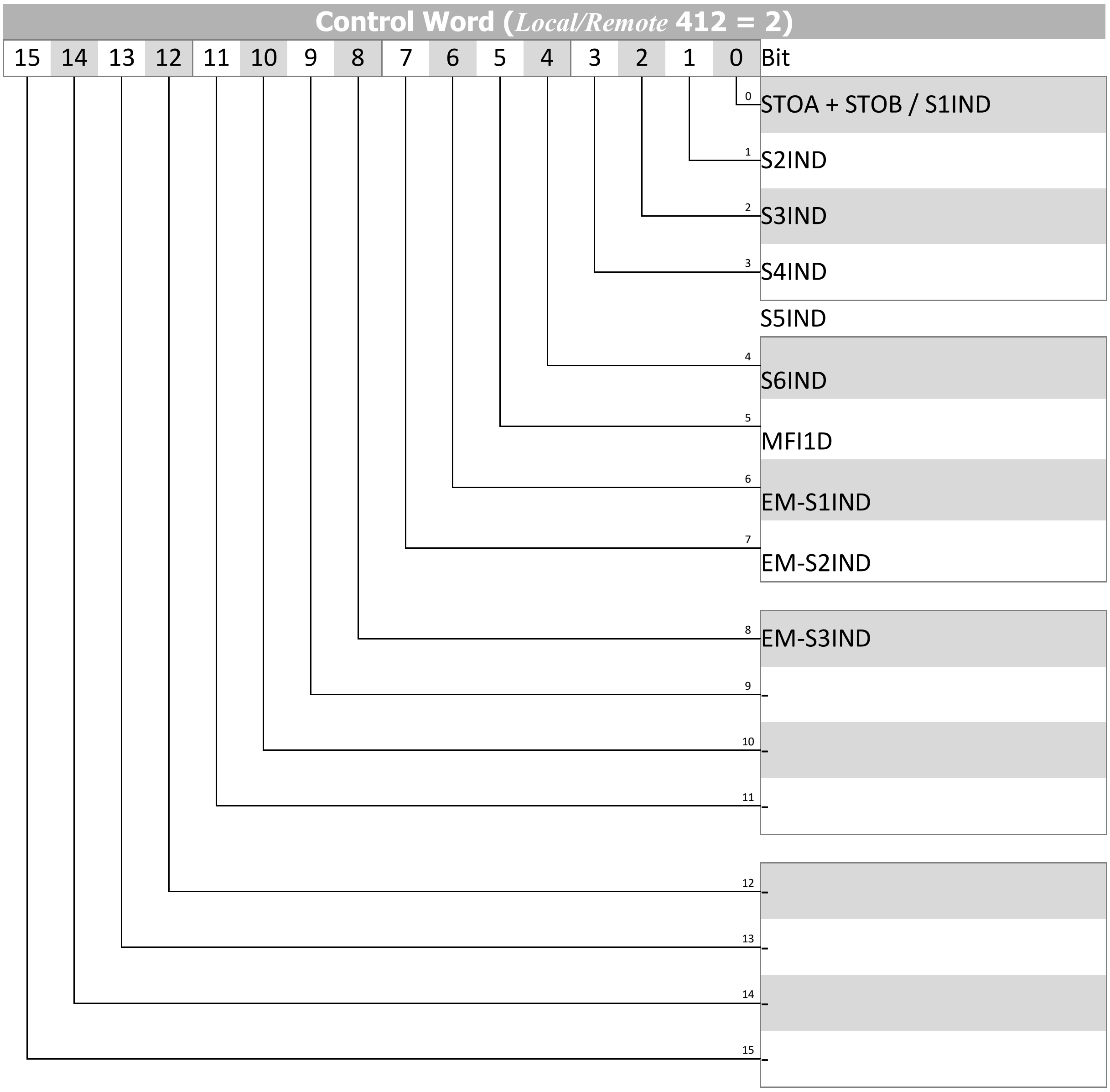
9.1 Control via Contacts / Remote-Contacts

In the operation mode "Control via Contacts" (*Local/Remote* **412** = 0), the frequency inverter is controlled via the digital inputs or via the multi-functional inputs that have been set to digital inputs.

In the operation mode "Control via Remote Contacts" (Parameter *Local/Remote* **412** = 2), the frequency inverter is controlled via the individual bits of the virtual digital inputs in the Control Word.

If the frequency inverter is controlled via the digital inputs, then in this operation mode control via the *Control Word* **410** does not apply.

Control Word (*Local/Remote* 412 = 2)



Bit	
0	STOA + STOB / S1IND
1	S2IND
2	S3IND
3	S4IND
	S5IND
4	S6IND
5	MFI1D
6	EM-S1IND
7	EM-S2IND
8	EM-S3IND
9	–
10	–
11	–
12	–
13	–
14	–
15	–

ACT: If the operation mode "Control via Remote Contacts" is used, then the Controller Release "S1IND" must be switched on and bit 0 of the Control Word must be set, in order to start the drive.

ACU: If the operation mode "Control via Remote Contacts" is used, then the Controller Release "STOA+STOB" must be switched on and bit 0 of the Control Word must be set, in order to start the drive.

With the use of Remote Contacts the signal sources (digital inputs or multifunctional inputs set as digital inputs) are taken virtually from the *Control Word* **410**. Signals at the hardware terminals are not evaluated in the standard operation modes (e.g. 72 - IN2D).

Operation modes which are marked with the extension "(Hardware") are available in order to evaluate signals at the hardware terminals.

Exception The release must always be made as follows:

- **ACT**: Hardware-Inputs S1IND (Terminal X11.3) and bit 0 "S1IND" of the Control Word.
- **ACU**: Hardware-Inputs STOA (Terminal X11.3) and STOB (Terminal X13.3) and bit 0 "STOA+STOB" of the Control Word

A Controller Release by software alone is not possible.

Parameter *Status Word* **411** has a length of 16 bits. The set bits have the following meaning:

Status Word	
Bit	
0	Ready to Switch On
1	Switched On
2	Operation enabled
3	Error
4	Voltage enabled
5	Quick Stop
6	Switch On disabled
7	Warning
8	-
9	Remote
10	Reference value reached
11	Limit value reached
12	-
13	-
14	-
15	Warning 2

Note:

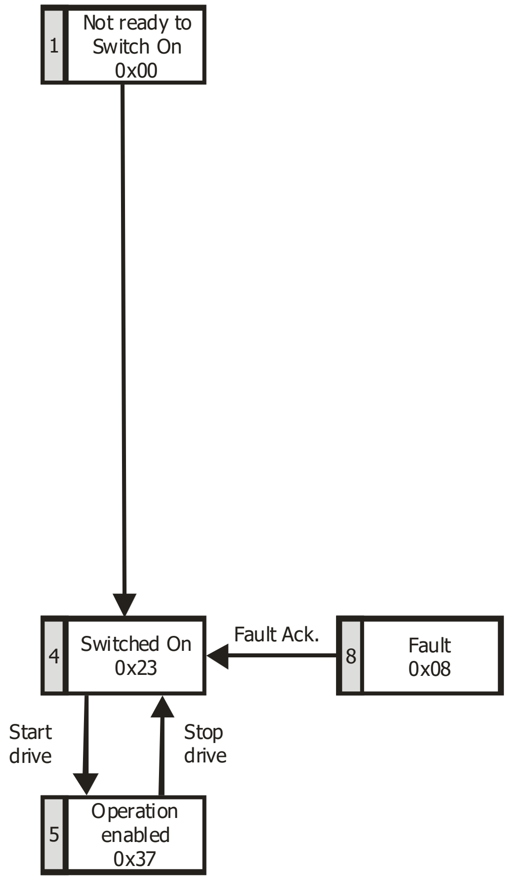
The frequency inverter supports an external 24 V voltage supply for the control electronics of the frequency inverter. Communication between the controlling device (PLC) and the frequency inverter is still possible even when the mains supply has been switched off.

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.

9.1.1 Device 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

"x" means any value.

If a fault has occurred, the cause of the fault can be read out via parameter *Current Error* **260**.

Status Word Bits 7 to 15:

Bit 7 "**Warning**" can signal an internal warning and results in the frequency inverter being switched off, depending on the cause. The evaluation of the warning is done by reading out the warning status via parameter *Warnings* **270**.

Bit 9, "**Remote**" is always set to "0" in the case of the control via contacts.

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.

9.2 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.

Control Word

Bit	
0	Switch On
1	Enable Voltage
2	Quick Stop
3	Enable Operation
4	no function
5	no function
6	no function
7	Reset fault
8	no function
9	no function
10	no function
11	no function
12	no function
13	no function
14	no function
15	no function

Status Word

Bit	
0	Ready to Switch On
1	Switched On
2	Operation enabled
3	Error
4	Voltage enabled
5	Quick Stop
6	Switch On disabled
7	Warning
8	no function
9	Remote
10	Reference value reached
11	Limit value reached
12	no function
13	no function
14	no function
15	Manufacturer dependent Warning 2

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

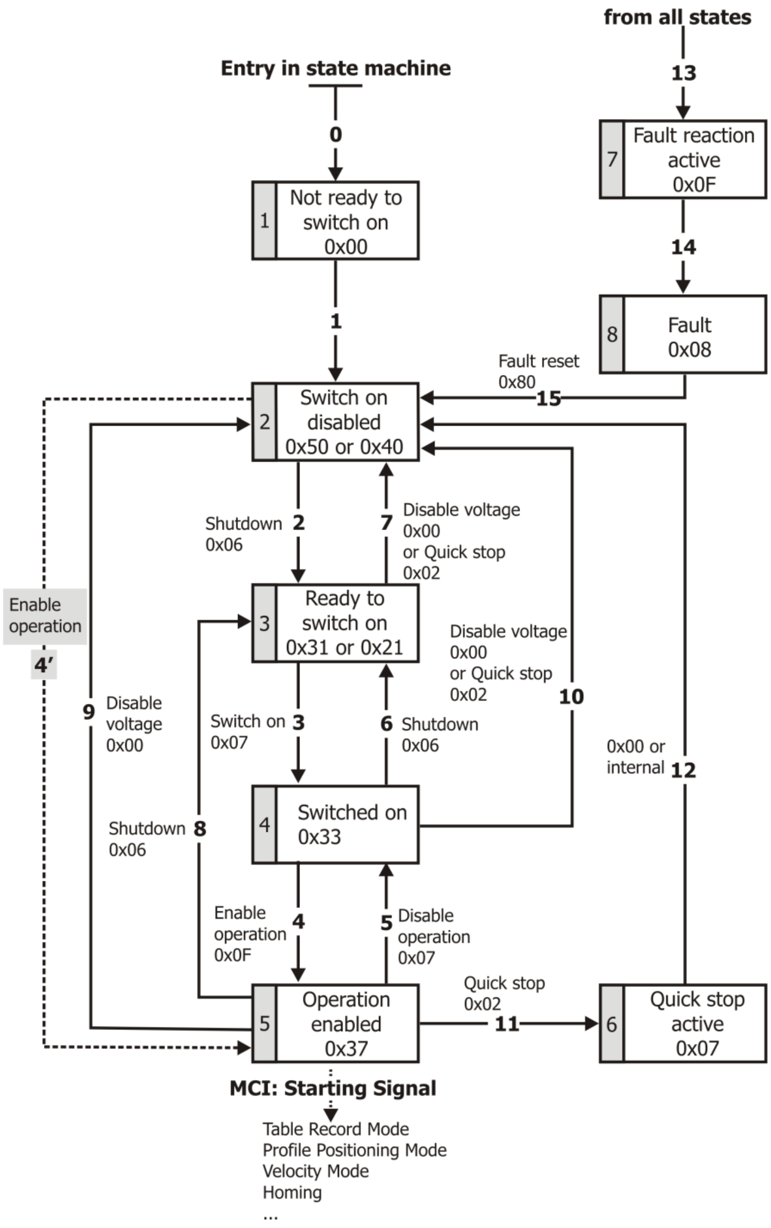
Bit 4 "Voltage enabled" = 0 signals "no mains supply" and that starting the drive is not possible.

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

Note:

The frequency inverter supports 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.

9.2.1 Statemachine diagram



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

Control Word						
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Command	Reset Fault	Enable Operation	Quick Stop	Enable Voltage	Switch On	Transitions
Shutdown	X	X	1	1	0	2, 6, 8
Switch On	X	0	1	1	1	3
Enable Operation	X	1	1	1	1	4
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
Reset Fault	0 ⇨ 1	x	x	x	x	15

"X" means any value.

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

The status word reflects the operation state.

Status Word						
	Bit 6	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0
State	Switch On disabled	Quick Stop	Error	Operation enabled	Switched On	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**
Error Reaction active	**0**	X	**1**	**1**	**1**	**1**
Error	**0**	X	**1**	**0**	**0**	**0**

"X" means any value.

Bit 7 "**Warning**" can be set at any time. It signals a device-internal warning. The active warning can be read out from the Warning Status with Parameter *Warnings* **270**.

Bit 9 "**Remote**" is set if the operation mode "Control via Statemachine" (*Local/Remote* **412** = 1) is set, and the Controller Release is switched on.

Logical combination of the digital control signals:

ACT: S1IND AND (Start Right OR Start Left).

ACU: STO = (STOA and STOB) AND (Start Right OR Start Left).

The frequency inverter can only be controlled, if the logical combination is true. The logical inputs for Start Right and Start Left can be directly linked to "On" or "Off" (Parameter *Start Clockwise* **68** and *Star Anticlockwise* **69**).

Bit 10 "**Reference frequency 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 (see frequency inverter operating instructions). For "Reference value reached" there is a hysteresis (tolerance range) which can be set via Parameter *Reference Value Reached: Hysteresis* **549** (see frequency inverter operating instructions).

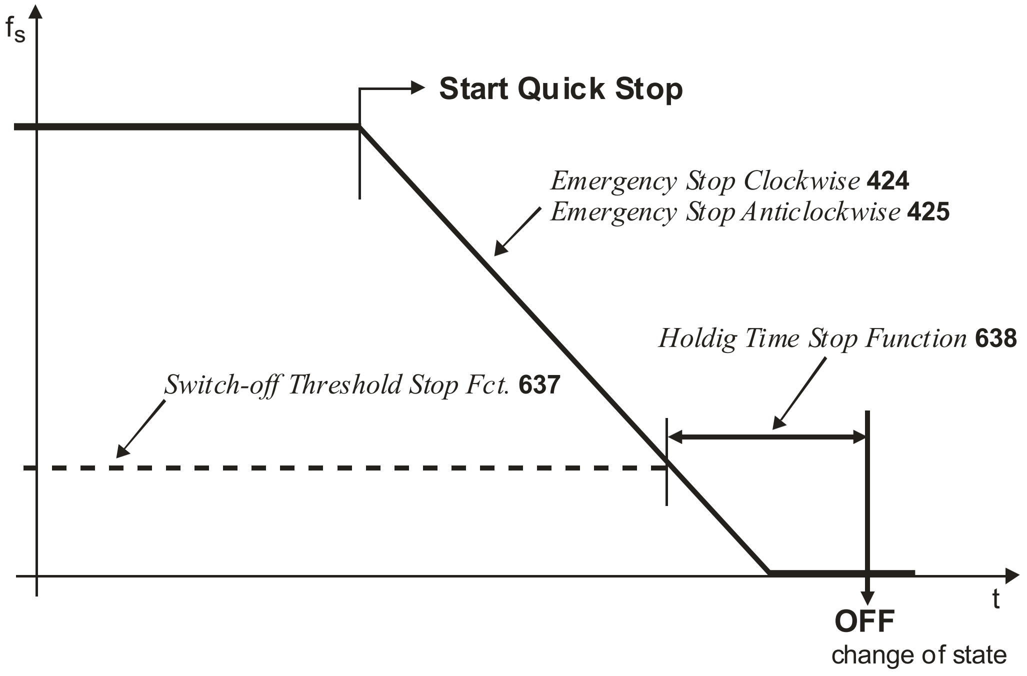
Bit 11 "**Limit value reached**" indicates that an internal limit is active. This may be the current limit, the torque limit or the overvoltage control. All functions result in the reference value being left or not reached.

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.

9.3 Behavior in Quick Stop

In this case, 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.

In a quick stop, the drive is brought to a standstill via the emergency stop ramps (*Emergency Stop Clockwise* **424** or *Emergency Stop Anticlockwise* **425**).


fs
Start Quick Stop
Emergency Stop Clockwise 424
Emergency Stop Anticlockwise 425
Holdig Time Stop Function 638
Switch-off Threshold Stop Fct. 637
t
OFF
change of state

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.

9.4 Behavior in State-Transition 5

- *392 state-transition 5*

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

Transition 5 392	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".

Setting 1 "DC-Brake" is only possible with applications with sensor-less control (e.g. configuration 110). Other configurations do not support this operation mode.

If the frequency inverter is operated with a configuration which does not support the operation mode DC-Brake (e.g. configuration 210, field-oriented control), value "1" cannot be used. In this case, the operation mode is not offered in the selection menus of the Operator Panel or the VPlus PC-Software.

The default value for *state-transition 5* **392** is operation mode 2 (ramp). For configurations with torque control, the default value is 0 (coasting).

If the configuration is changed, the value set for *state-transition 5* **392** is also changed, if necessary.

If *state-transition 5* **392** was triggered with value 1 "Direct current brake", a new Control Word will only be accepted after the transition process is complete. The change of state from "Operation released" to "Switched On" is carried out after the *Braking Time* **632** parameterized for the DC brake has elapsed.

If parameter *state-transition 5* **392** = 2 "ramp" is set, the Control Word can be set to 0x0F again, while the drive is decelerating. In this way, the drive accelerates to its set reference value again and remains in the state "Operation released".

The change of state from "Operation enabled" to "Switched On" is done after the value has dropped below the set switch-off threshold **and** the set holding time has elapsed (equivalent to the behavior in the case of a quick stop). In this case, 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.

9.5 Reference value

9.5.1 Reference frequency

The following signal sources are available for determining the reference frequency:

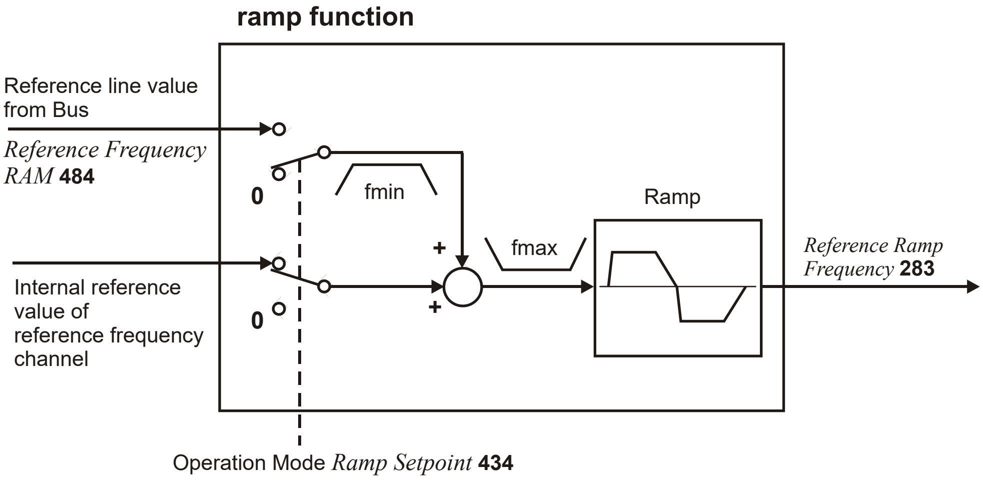
- External reference value as reference line value from bus
- Internal reference value from the reference frequency channel

The reference line value with the parameter name *Reference Frequency RAM* **484** is sent to the frequency inverter via the bus.

The internal reference value reaches the frequency inverter via the reference frequency channel. The reference frequency channel is configured via parameter *Reference Frequency Source* **475**.

The actual value can be read back vie different parameters. The selection depends on the control method used and the application. The following parameters are possible, for example:

– *Stator Frequency* **210**	(data type long, Hz)
– *Encoder 1 Frequency* **217** (Speed Sensor 1 Frequency)	(data type long, Hz)
– *Encoder 1 Speed* **218** (Speed Sensor 1 Speed)	(data type int, rpm).


ramp function
Reference line value from Bus
Reference Frequency RAM **484**
0
fmin
Ramp
+
fmax
Reference Ramp Frequency **283**
Internal reference value of reference frequency channel
+
0
Operation Mode *Ramp Setpoint* **434**

The internal reference value from the reference frequency channel and the reference line value (line setpoint) can be fed to the ramp individually or as an added variable. The result is available at the output of the ramp function as *Reference Ramp Frequency* **283**.

The operation mode of the ramp function is set via the data set related parameter *Ramp Setpoint* **434**.

For information on the reference frequency channel, refer to the frequency inverter operating instructions.

Ramp Setpoint 434		Function
1 –	Internal Setpoint	The internal reference frequency is determined from the reference frequency channel.
2 –	Line Setpoint	The reference value is supplied externally via the bus
3 –	Internal + Line Setpoint	Addition (considering the sign) of internal reference frequency and reference line value

For *Ramp Setpoint* **434** = 3, the sign of the overall reference value is obtained from the addition of the reference frequency + the reference line value.

Note:

If *Ramp Setpoint* **434** = 2 (reference line value only), this reference line value is limited to fmin.

In this case, consider that the sign in front of fmin with reference value = 0 is derived from the sign in front of the last reference line value which was not 0.

After Mains On, the reference line value is limited to +fmin (motor clockwise).

The reference values can be controlled at the frequency inverter via the control unit KP500 or the control software VPlus via the following parameters:

Internal Reference Frequency **228**	= internal ref. value from ref. freq. channel
Reference Bus Frequency **282**	= reference line value from serial interface
Reference Ramp Frequency **283**	= sum of internal reference frequency + reference line value

Note:

If the serial interface of the communication module CM-232 / CM-485 is connected to a PLC, for example, VPlus cannot access the frequency inverter via this interface directly.

In this case, the PC must be connected via the optional interface adapter KP232.

For information on the reference frequency channel, refer to the frequency inverter operating instructions.

9.5.2 Reference percentage value

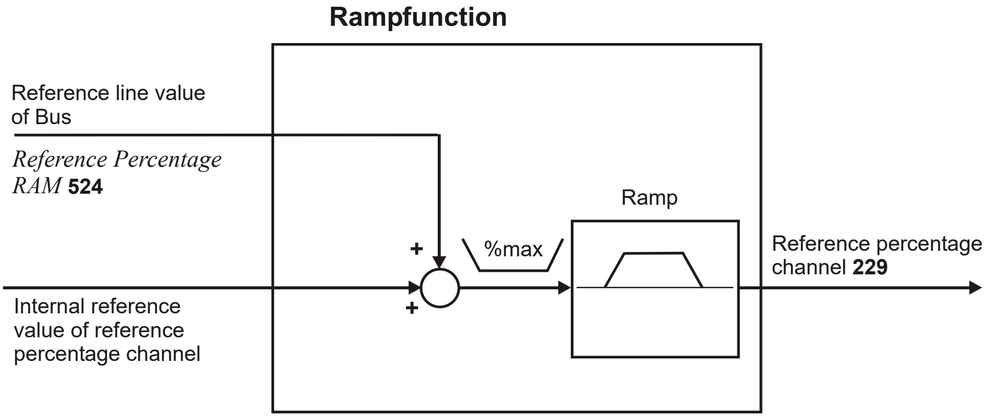
The following signal sources are available for determining the reference percentage:

- External reference value as reference line value from bus
- Internal reference value from the reference percentage channel

The reference line value with the parameter name *Reference Percentage RAM* **524** is sent to the frequency inverter via the bus.

The internal reference value reaches the frequency inverter via the reference percentage channel. The reference percentage channel is configured via parameter *Reference Percentage Source* **476**.

The parameter *Reference Percentage Value* **229** can be read back as an actual value


Rampfunction
Reference line value of Bus
Reference Percentage RAM 524
+
%max
Ramp
Reference percentage channel 229
+
Internal reference value of reference percentage channel

The internal reference value from the reference percentage channel and the reference line value are fed to the ramp as an added variable. The sum of the two variables has a positive or a negative sign according to the mathematical operation.

For information on the reference percentage channel, refer to the frequency inverter operating instructions.

Note:

The actual value parameter *Reference percentage* **229** can be checked at the frequency inverter via the control unit KP500 or the control software VPlus.

For example, the *Reference Percentage Value* **229** is processed in the frequency inverter via the functions "Technology Controller" or "Torque Controller" (e.g. speed controller for torque limitation. For more information refer to the frequency inverter operating instructions.

Note:

If the serial interface of the module CM-232/CM-485 is connected to a PLC, for example, VPlus cannot access the frequency inverter via this interface directly.

In this case, the PC must be connected via the optional interface adapter KP232.

10 Actual Values

Actual Values		
No.	**Description**	**Function**
11	VABus SST-Error-Register	Modbus or VABus Error Register. See Chapter 7.3.9 "Exception Codes".
282	Reference Bus Frequency	Reference value from the serial interface.
283	Reference Ramp Frequency	Reference value from the Reference Frequency Channel.
411	Status Word	Modbus or VABus Status Word. See Chapter 10 "Control / Reference Value"

11 Parameter List

▤ The parameter is available in the four data sets.

11.1 Actual Values ("Actual" Menu)

Actual Value Parameters				
No.	**Description**	**Units**	**Value Range**	**Chapter**
RS485/RS232				
11	VABus SST-Error-Register	-	0 ... 15	6.3.9
Actual Frequency Inverter Values				
249	Active Data Set	-	0 ... 4	9
260	Actual Error	-	0 ... 0xFFFF	12.3
270	Warnings	-	0 ... 0xFFFF	12.1
274	Application Warnings	-	0 ... 0xFFFF	12.2
282	Reference Bus Frequency	Hz	-999,99 ... 999,99	10
283	Reference Ramp Frequency	Hz	-999,99 ... 999,99	10
Bus Control				
411	Status Word	-	0 ... 0xFFFF	9

Note:
The parameters *Current error* **260**, *Warnings* **270** and *Application Warnings* **274** are only accessible via the manufacturer objects 0x2nnn. They cannot be accessed via the VPlus program or the KP500 control unit.

11.2 Parameters ("Para" Menu)

Parameter				
No.	**Description**	**Units**	**Value Range**	**Chapter**
RS485/RS232				
10	Baud rate	-	1…6	5.2.1
Bus Control				
392	state-transition 5	-	Selection	9.4
RS485/RS232				
395	Protocol (CM / X21)	-	Selection	5.2.4
Bus Control				
410	Control Word	-	0 … 0xFFFF	9
▤ 412	Local/Remote	-	Selection	9
RS485/RS232				
413	RS232/RS485 Watchdog Timer	s	0…10000	5.2.5
Bus Control				
414	Data Set Selection	-	0 … 4	9
Fixed Frequency Values				
484	Reference Frequency RAM	Hz	-999,99 … 999,99	9
Fixed Percentage Values				
524	Reference Percentage RAM	%	-300,00 … 300,00	9
Modbus (RTU/ASCII)				
1375	Modbus Parity	-	Selection	5.2.2
1376	Modbus Address	-	1 … 247	5.2.3

12 Annex

12.1 Warning Messages

The various control functions and methods as well as the hardware of the frequency inverter contain functions that continuously monitor the application. In addition to the messages documented in the manual, the following warning messages are activated by the communication module CM-232 or CM-485.

The warning messages are given via parameter *Warnings* **270**, bit-coded according to the following scheme. The parameter *Warnings* **270** is meant to be read out by PLCs, parameter *Warnings* **269** shows the same information with a short text description in VPlus and Keypad KP500.

Warning Messages

Bit-No.	Warning Code	Description	
0	0x0001	Warning Ixt	1) 2)
1	0x0002	Warning Short Time Ixt	1)
2	0x0004	Warning Long Time Ixt	2)
3	0x0008	Warning Heat Sink Temperature Tk	
4	0x0010	Warning Inside Temperature Ti	
5	0x0020	Warning I-Limit	
6	0x0040	Warning Init	
7	0x0080	Warning Motor Temperature	
8	0x0100	Warning Mains Phase 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 Slave in Fault	
14	0x4000	Warning Udc	
15	0x8000	ACT: Warning V-belt ACU: Warning Application	

1) 2): Bit 0 "Warning Ixt" is set,

- if Bit 1 "Warning Short Term Ixt" or
- if Bit 2 "Warning Long Term Ixt" is set.

In Parameter *Warnings* **270** several warnings can be set at the same time.

Example:

Message	Warning Code	Comment
Warning Ixt	0x0001	Set on Short Term or Long Term Ixt.
Short Term Ixt	0x0002	
Warning Limit Heat Sink Temperature	0x0008	
Warning Limit Motor Temperature	0x0080	
Sum	**0x008B**	

12.2 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 Modbus.

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

Note:

The functionality “Warning messages Application” is included in the device series ACU, in the device series ACT this functionality is not included.

The meaning of the individual warnings are described in detail in the operating instructions ACU.

12.3 Error Messages

VABus		
F20	10	Watchdog RS232/RS485-Connection. Communication error according to Parameter *RS232/RS485s Watchdog Timer* **413**.

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.

Index

Bonfiglioli Worldwide Locations

Australia

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

Brazil

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

China

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

France

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

Germany

Bonfiglioli Deutschland GmbH
Sperberweg 12 - 41468 Neuss
Tel. +49 0 2131 2988 0
🔧 | 🛒 | 🛠

Bonfiglioli Vectron GmbH
Europark Fichtenhain B6 - 47807 Krefeld
Tel. +49 0 2151 8396 0
⚙ | 🔧 | 🛒 | 🛠

O&K Antriebstechnik GmbH
Ruhrallee 8-12 - 45525 Hattingen
Tel. +49 0 2324 2050 1
⚙ | 🔧 | 🛒 | 🛠

⚙ PRODUCTION
🔧 ASSEMBLY
🛒 SALES
🛠 SERVICE

India

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

Discrete Manufacturing & Process Industries - Mechatronic & Motion
Survey No. 528/1
Perambakkam High Road Mannur Village,
Sriperumbudur Taluk Chennai - 602 105
Tel. +91 844 844 8649
🔧 | 🛠

Discrete Manufacturing & Process Industries
Plot No.A-9/5, Phase IV MIDC Chakan,
Village Nighoje Pune - 410 501
Tel. +91 844 844 8649
🔧 | 🛠

Italy

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

Mobility & Wind Industries
Via Enrico Mattei, 12 Z.I. Villa Selva
47100 Forlì
Tel. +39 0543 789111
⚙ | 🔧 | 🛒 | 🛠

Discrete Manufacturing & Process Industries
Via Sandro Pertini lotto 7b
20080 Carpiano
Tel. +39 02985081
🔧 | 🛒 | 🛠

Bonfiglioli Mechatronic Research S.p.A
Via Unione 49 - 38068 Rovereto
Tel. +39 0464 443435/36
⚙ | 🔧 | 🛒 | 🛠

New Zealand

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

Singapore

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

Slovakia

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

South Africa

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

Spain

Tecnotrans Bonfiglioli S.A
Pol. Ind. Zona Franca, Sector C,
Calle F, n° 6 - 08040 Barcelona
Tel. +34 93 447 84 00
🔧 | 🛒 | 🛠

Turkey

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

United Kingdom

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

USA

Bonfiglioli USA Inc.
3541 Hargrave Drive
Hebron, Kentucky 41048
Tel. +1 859 334 3333
⚙ | 🔧 | 🛒 | 🛠

Vietnam

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

Bonfiglioli

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

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

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

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

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

COD. VEC 383 R4

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

tel: +39 051647 3111
www.bonfiglioli.com