Drive Technology \ Drive Automation \ System Integration \ Services

SEW EURODRIVE

Manual

Fieldbus Gateway UFR41B
EtherNet/IP, Modbus/TCP and PROFINET IO

Edition 05/2009

16798813 / EN

1 General Information

1.1 How to use the manual

The manual is part of the product and contains important information on operation and service. The manual is written for all employees who assemble, install, startup, and service the product.

The manual must be accessible and legible. Make sure that persons responsible for the system and its operation, as well as persons who work independently on the unit, have read through the manual carefully and understood it. If you are unclear about any of the information in this documentation, or if you require further information, contact SEW-EURODRIVE.

1.2 Structure of the safety notes

The safety notes in this manual are structured as follows:

Pictogram	⚠ **SIGNAL WORD**
⚠	Type and source of danger. Possible consequence(s) if the safety notes are disregarded. • Measure(s) to prevent the danger.

Pictogram	**Signal word**	**Meaning**	**Consequences if disregarded**
Example: ⚠ General danger ⚡ Specific danger, e.g. electric shock	⚠ **DANGER**	Imminent danger	Severe or fatal injuries
	⚠ **WARNING**	Possible dangerous situation	Severe or fatal injuries
	⚠ **CAUTION**	Possible dangerous situation	Minor injuries
	NOTICE	Possible damage to property	Damage to the drive system or its environment
ℹ	**TIP**	Useful information or tip. Simplifies the handling of the drive system.	

1.3 Rights to claim under limited warranty

A requirement of fault-free operation and fulfillment of any rights to claim under limited warranty is that you adhere to the information in the manual. Therefore, read the manual before you start operating the device.

1.4 Exclusion of liability

You must comply with the information in the manual and the documentation of the units connected to the fieldbus gateway to ensure safe operation and to achieve the specified product characteristics and performance features. SEW-EURODRIVE assumes no liability for injury to persons or damage to equipment or property resulting from non-observance of the operating instructions. In such cases, any liability for defects is excluded.

1.5 Copyright notice

2 Safety Notes

2.1 Other applicable documentation

- Installation and startup may be carried out only by trained personnel observing the relevant accident prevention regulations and the following documents:
  - "MOVIDRIVE® MDX60B/61B" operating instructions
  - "MOVITRAC® B" operating instructions
  - "MOVIAXIS®" operating instructions
- Read through these documents carefully before you commence installation and startup of the UFR41B fieldbus gateway.
- As a prerequisite to fault-free operation and fulfillment of warranty claims, you must adhere to the information in the documentation.

2.2 General safety notes for bus systems

This communication system lets you adjust inverters and servo inverters to a variety of different applications. As with all bus systems, there is a danger of invisible, external (as far as the inverter is concerned) modifications to the parameters which give rise to changes in the unit behavior. This may result in unexpected (not uncontrolled) system behavior.

2.3 Safety functions

The inverters and servo drives are not allowed to perform any safety functions unless they are subordinate to other safety systems. Use higher-level safety systems to ensure protection of equipment and personnel.

For safety applications, ensure that the information in the following publications is observed: "Safe Disconnection for MOVIDRIVE® B / MOVITRAC® B / MOVIAXIS®".

2.4 Hoist applications

MOVIDRIVE® MDX60B/61B, MOVITRAC® B and MOVIAXIS® must not be used as a safety device in hoist applications.

Use monitoring systems or mechanical protection devices as safety equipment to avoid possible damage to property or injury to people.

2.5 Product names and trademarks

The brands and product names contained within this manual are trademarks or registered trademarks of the titleholders.

2.6 Waste disposal

Observe the applicable national regulations.

Dispose of the following materials separately in accordance with the country-specific regulations in force, as:

- Electronics scrap
- Plastic
- Sheet metal
- Copper

3 Introduction

3.1 Content of the manual

This user manual describes how to:

- Connect the UFR41B fieldbus gateway to MOVIDRIVE® B, MOVITRAC® B inverters and to the MOVIAXIS® servo inverter.
- Startup MOVIDRIVE® B, MOVITRAC® B and MOVIAXIS® for gateway operation.
- Startup the fieldbus gateway UFR41B on the fieldbus system EtherNet/IP, Modbus/TCP and PROFINET IO.
- Configure the EtherNet/IP master with EDS files.
- Configure the Modbus/TCP master.
- Configure the PROFINET IO master using GSD files.

3.2 Characteristics

The powerful, universal fieldbus interfaces of the UFR41B fieldbus gateway enable you to use the option to connect to higher-level automation systems via EtherNet/IP, Modbus/TCP and PROFINET IO.

3.2.1 Process data exchange

The UFR41B fieldbus gateway allows for digital access to most parameters and functions via EthernetNet/IP, Modbus/TCP, and PROFINET IO interfaces. Control is performed via fast, cyclic process data. Via this process data channel, you can enter setpoints and trigger various control functions, such as enable, normal stop, rapid stop, etc. At the same time you can also use this channel to read back actual values, such as actual speed, current, unit status, error number or reference signals.

3.2.2 Parameter access

In EtherNet/IP operation, the parameters of the inverter are set solely via *explicit messages*.

In Modbus/TCP operation, the controller can access the parameters via the 8 byte MOVILINK® parameter channel.

In PROFINET operation, two parameter access options are available:

- The PROFIDRIVE data record 47 offers access to all unit information also in PROFINET operation
- The parameter mechanism offers universal access to any unit information

3.2.3 Monitoring functions

Using a fieldbus system requires additional monitoring functions, for example, time monitoring of the fieldbus (fieldbus timeout) or rapid stop concepts. You can determine, for instance, which fault responses should be triggered in the event of a bus error. The parameters for the fault response can be set in the servo inverter / inverter. A rapid stop is useful for many applications. This is why the fieldbus gateway will stop the lower-level drives in the event of a fieldbus timeout. As the range of functions for the control terminals is also guaranteed in fieldbus mode, you can continue to implement rapid stop concepts using the servo inverters/inverters connected to the fieldbus gateway.

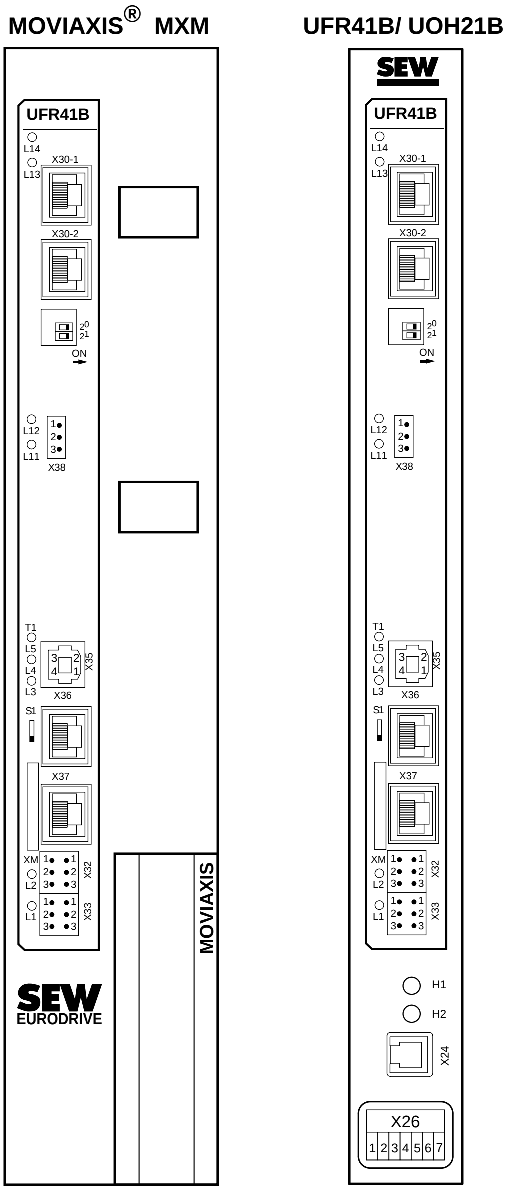
4 Assembly and Installation Instructions

This chapter contains information on the assembly and installation of the UFR41B fieldbus gateway in a MOVIAXIS® master module MXM or in an UOH21B gateway housing.

4.1 Installation options of the UFR41B fieldbus gateway

Observe the following installation instructions:

	TIP
i	Only SEW-EURODRIVE is allowed to install/remove the UFR41B fieldbus gateway into/from a MOVIAXIS® master module MXM and an UOH21B gateway housing.



MOVIAXIS® MXM **UFR41B/ UOH21B**

65055AXX

4.2 Voltage supply

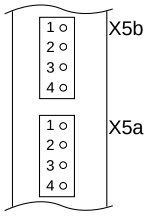
Voltage supply, system bus and fieldbus interfaces as well as the engineering interface are located at different potential levels (see chapter 13.1).

4.2.1 Voltage supply in the MOVIAXIS® master module

TIP

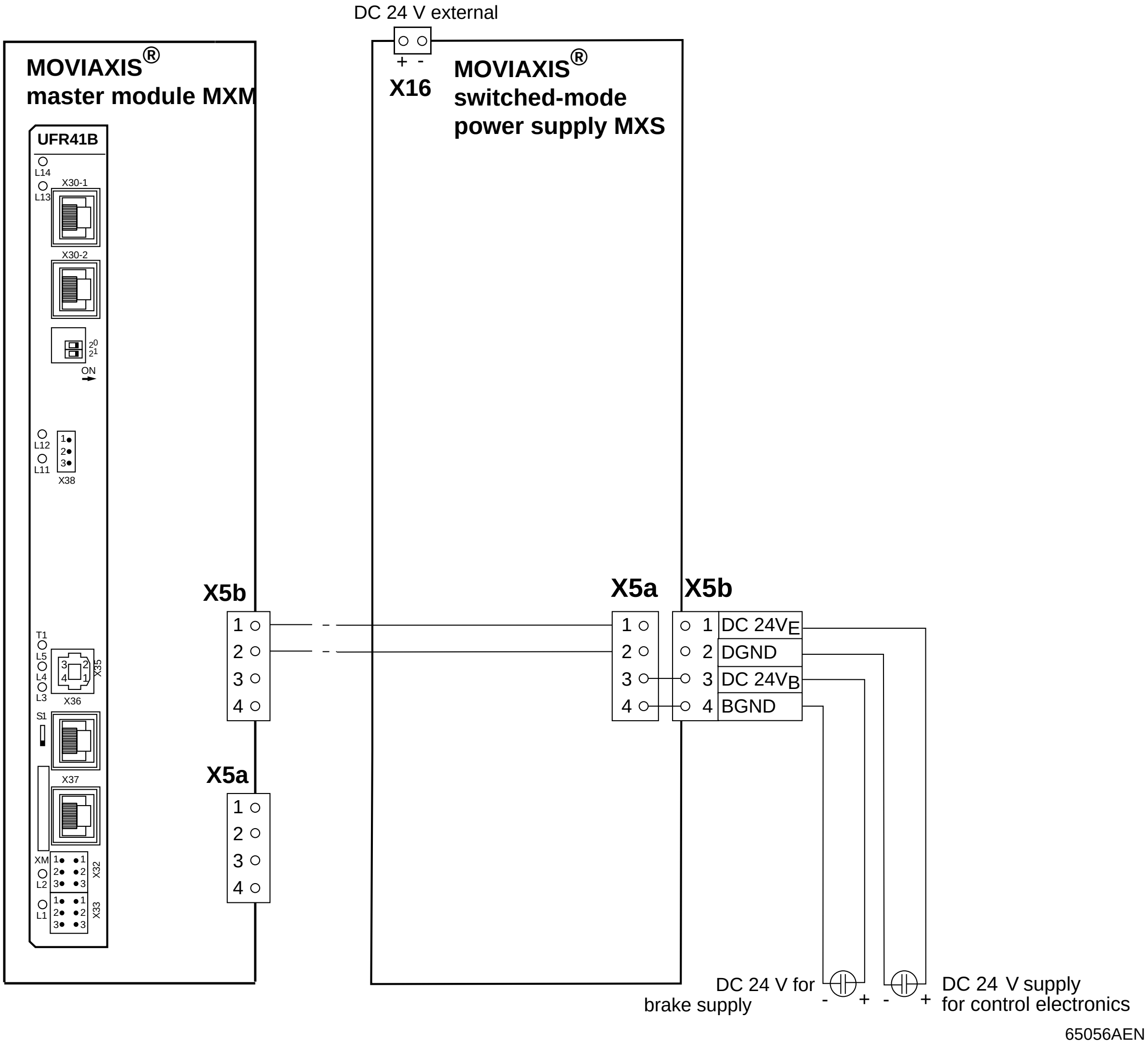
The MOVIAXIS® master module MXM provides additional connections, which are described in the following section.

Functional description of the terminals, X5a/X5b (MOVIAXIS® master module)

MOVIAXIS® master module MXM	Designation	Terminal		Function
 59233AXX	**X5b connector**	**X5b:1** **X5b:2** **X5b:3** **X5b:4**	DC 24 V_E DGND DC 24 V_B BGND	Voltage supply for control electronics Reference potential for control electronics Voltage supply for brake Reference potential for brake connection
	X5a connector	**X5a:1** **X5a:2** **X5a:3** **X5a:4**	DC 24 V_E DGND DC 24 V_B BGND	Voltage supply for control electronics Reference potential for control electronics Voltage supply for brake Reference potential for brake connection

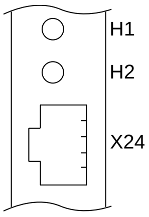
- The X5a and X5b connectors are connected in parallel. In this way, the voltage supply of the MOVIAXIS® master module can be provided from the right to X5b or from below to X5a. With connection to X5a, further modules can be connected via X5b (e.g. supply module, axis module). The voltage supply for the brake (X5a/b:3, 4) is fed through the MOVIAXIS® master module.
- The UFR41B fieldbus gateway can be supplied from the MOVIAXIS® switched-mode power supply (MXS) or from an external voltage source. To do so, connect X5 between the individual units.
- If the UFR41B fieldbus gateway is connected with DC 24 V from the MOVIAXIS® switched-mode power supply, the functioning of the option is maintained after disconnection from the power supply. This is the case if the DC link voltage is maintained or an external DC 24 V supply is present from the MOVIAXIS® switched-mode power supply.

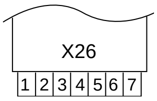
Wiring diagram



4.2.2 Voltage supply in the UOH21B gateway housing

Description of the terminals and LED functions

Front view MOVITRAC® B / compact controller	**Designation**	**LED Terminal**		**Function**
 58905AXX	**LED**	**H1** **H2**		Reserved Reserved
	X24 connector: RJ10 socket	**X24:4** **X24:3** **X24:2** **X24:1**		**No function.** Engineering cannot be performed using X24.

Side view Compact controller	**Designation**	**Terminal**		**Function**
 58906AXX	**X26 connector: CAN 1 and voltage supply (plug-in terminal)**	**X26:1** **X24:2** **X24:3** **X24:4** **X26:5** **X26:6** **X26:7**	CAN1H CAN1L DGND Reserved Reserved DGND DC 24 V	System bus CAN 1 high System bus CAN 1 low Reference potential control/CAN1 - - Reference potential for UFx41B Voltage supply for controller

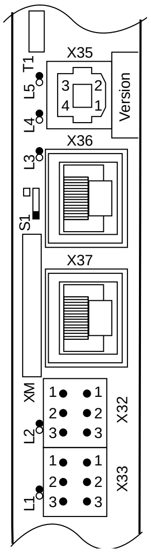
Connection of CAN 1 system bus / voltage supply (X26 connector)

The connections for CAN 1 (X26:1/2/3 and connector X33) are connected in parallel. The UFR41B fieldbus gateway is supplied with voltage in the UOH21B gateway housing via X26:6/7.

4.3 Connecting inverters and engineering-PC

4.3.1 Functional description of the terminals, DIP switches and LED of the UFR41B option

Connectors, LEDs and DIP switches in the upper area of the UFR41B fieldbus gateway allow for connection to EtherNet/IP (see chapter "Connecting the UFR41B fieldbus system to an EtherNet/IP network"), Modbus/TCP (see chapter "Connecting the UFR41B fieldbus to a Modbus/TCP network") and PROFINET IO fieldbus systems (see chapter "Connecting the UFR41B fieldbus gateway to a PROFINET IO network")

Front view UFR41B fieldbus gateway	**Designation**	**LED DIP switch Terminal**		**Function**
UFF41B UFR41B  X35 T1 L5 L4 3 2 4 1 Version X36 L3 S1 X37 XM L2 X32 L1 X33 1 2 3 64418AXX	**LED**	**LED 1** **LED 2** **LED 3** **LED 4** **LED 5**	CAN 1 status CAN 2 status Program status Gateway status Gateway error	Status of CAN 1 system bus Status of CAN 2 system bus Status of gateway program Status of gateway firmware Status of gateway error (see section "Error messages of the fieldbus gateway")
	X35 connector: USB connection	**X35:1** **X35:2** **X35:3** **X35:4**	USB+5 V USB- USB+ DGND	DC 5 V voltage supply USB- signal USB+ signal Reference potential
	X36 connector: Connection of an EtherCAT based system bus (RJ45 socket)	**X36**	Standard Ethernet assignment	System bus SBUS^{plus} (in preparation)
	X37 connector: Ethernet connection (RJ45 socket)	**X37**		Ethernet for engineering
	X32 connector: System bus CAN 2 (electrically isolated) (plug-in terminals)	**X32:1** **X32:2** **X32:3**	BZG_CAN 2 CAN 2H CAN 2L	Reference potential for system bus CAN 2 System bus CAN 2 high System bus CAN 2 low
	X33 connector: System bus CAN 1 (plug-in terminals)	**X33:1** **X33:2** **X33:3**	DGND CAN 1H CAN 1L	Reference potential for system bus CAN 1 System bus CAN 1 high System bus CAN 1 low
	DIP switch	**S1**	Top Bottom	Default IP address (192.168.10.4) IP parameter from SD memory card
	Memory card	**M1**		Memory for firmware, gateway application, gateway configuration, and inverter parameters
	Button	**T1**		For Bootloader update (see section "SD memory card OMG4.B")

4.3.2 Connecting CAN 1 system bus (X33 connector)/ CAN 2 (X32 connector)

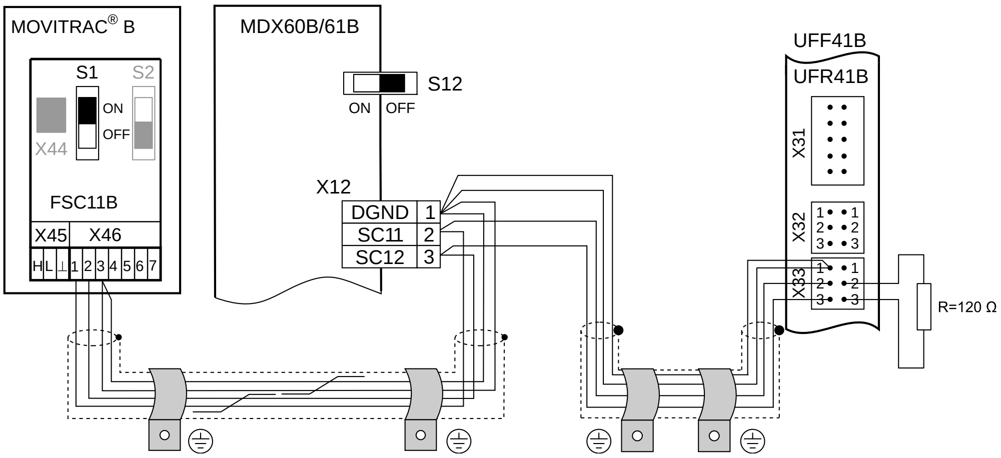
Do not connect more than 16 units to the CAN 1 or CAN 2 system bus in gateway operation.

TIPS

- The CAN 1 system bus is **not** electrically isolated. Therefore, it is recommended to use the CAN 1(X33 or X26 with UFR41B/UOH21B) interface to connect inverters via the system bus in the control cabinet. Set the *P881 SBus address* parameter in increasing order to values 1 - 16 if the slave unit is connected to CAN 1 or the fieldbus gateway.
- The CAN 2 system bus is electrically isolated. Therefore, preferably use interface CAN 2 (X32) for connecting field units or units in other control cabinets. Set the *P881 SBus address* parameter in increasing order to values 17 - 34 if the unit is connected to CAN 2 or the fieldbus gateway.

The CAN system bus supports transmission systems compliant with ISO 11898. For detailed information on the CAN system bus, refer to the "MOVIDRIVE® Communication and Fieldbus Device Profile" manual. You can order this manual from SEW-EURODRIVE.

Wiring diagram for MOVIDRIVE® B, MOVITRAC® B on CAN 1 system bus



64714AXX

Cable specification

- Use a 2 x 2-core twisted and shielded copper cable (data transmission cable with braided copper shield). Clamping without conductor end sleeves is possible in accordance with IEC 60999. The cable must meet the following specifications:
  - Cable cross-section 0.2 to 1.0 mm^2 (AWG 24 - AWG 18)
  - Cable resistance 120 Ω at 1 MHz
  - Capacitance per unit length = 40 pF/m at 1 kHz

  Suitable cables include CAN bus or DeviceNet cables.

Cable length

- The permitted total cable length depends on the baud rate setting of the system bus:

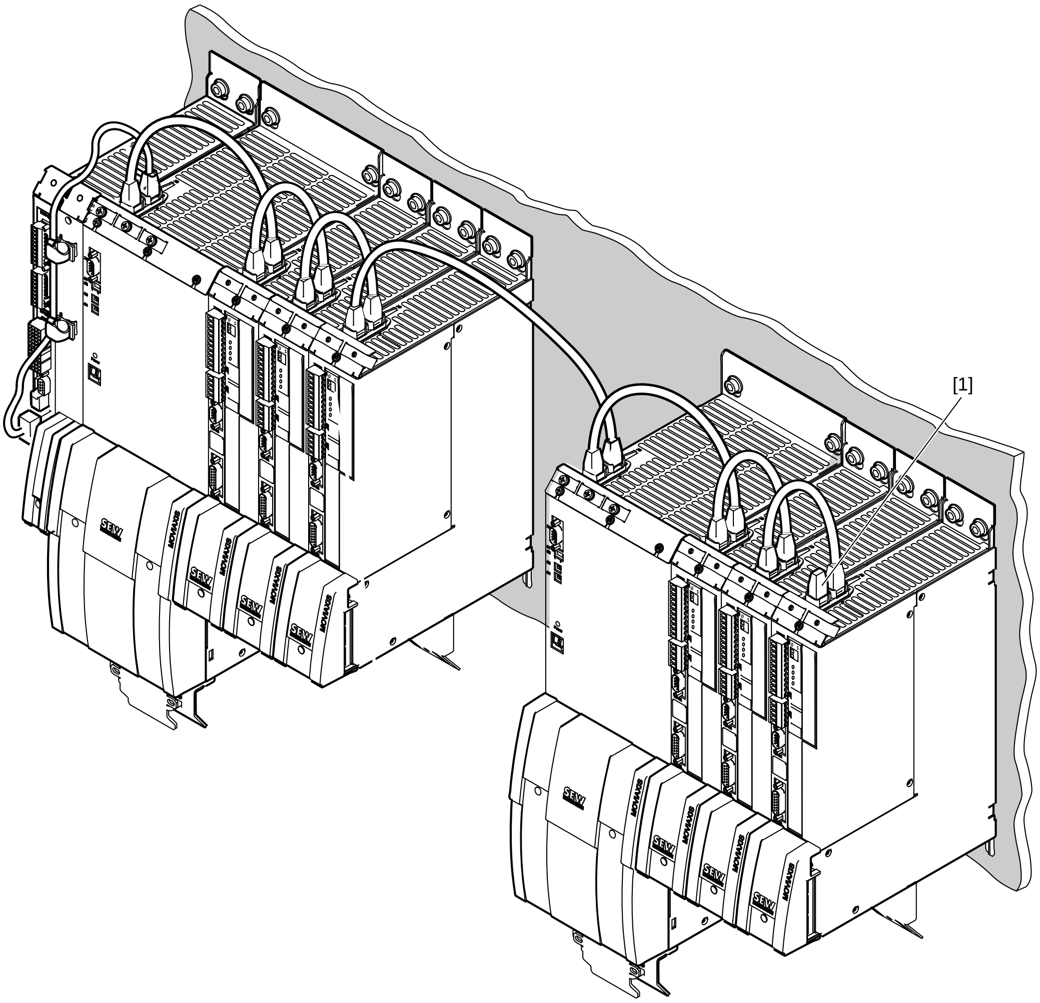
– 125 kBd	→	500 m
– 250 kBd	→	250 m
– **500 kBaud**	→	**100 m**
– 1000 kBd	→	40 m

Terminating resistor

- Switch on the system bus terminating resistor at the start and end of the CAN system bus connection (MOVIDRIVE® B, DIP switch S12 = ON; MOVITRAC® B, DIP switch S1 = ON). For all other devices, switch off the terminating resistor (MOVIDRIVE® B, DIP switch S12 = OFF; MOVITRAC® B, DIP switch S1 = OFF). If the fieldbus gateway is, for example, located at the end of the CAN 2 system bus, you have to connect a terminating resistor of 120 Ω between pins X32:2 and X32:3 (for CAN 1: terminating resistor between pins X33:2 and X33:23).

	CAUTION
⚠	• There **must not** be any potential displacement between the units connected via the CAN 2 system bus. • There **must not** be any potential displacement between the units connected via the CAN 1 system bus. • Take suitable measures to avoid potential displacement, such as connecting the unit ground connectors using a separate cable.

Wiring diagram for MOVIAXIS® on CAN 1 system bus



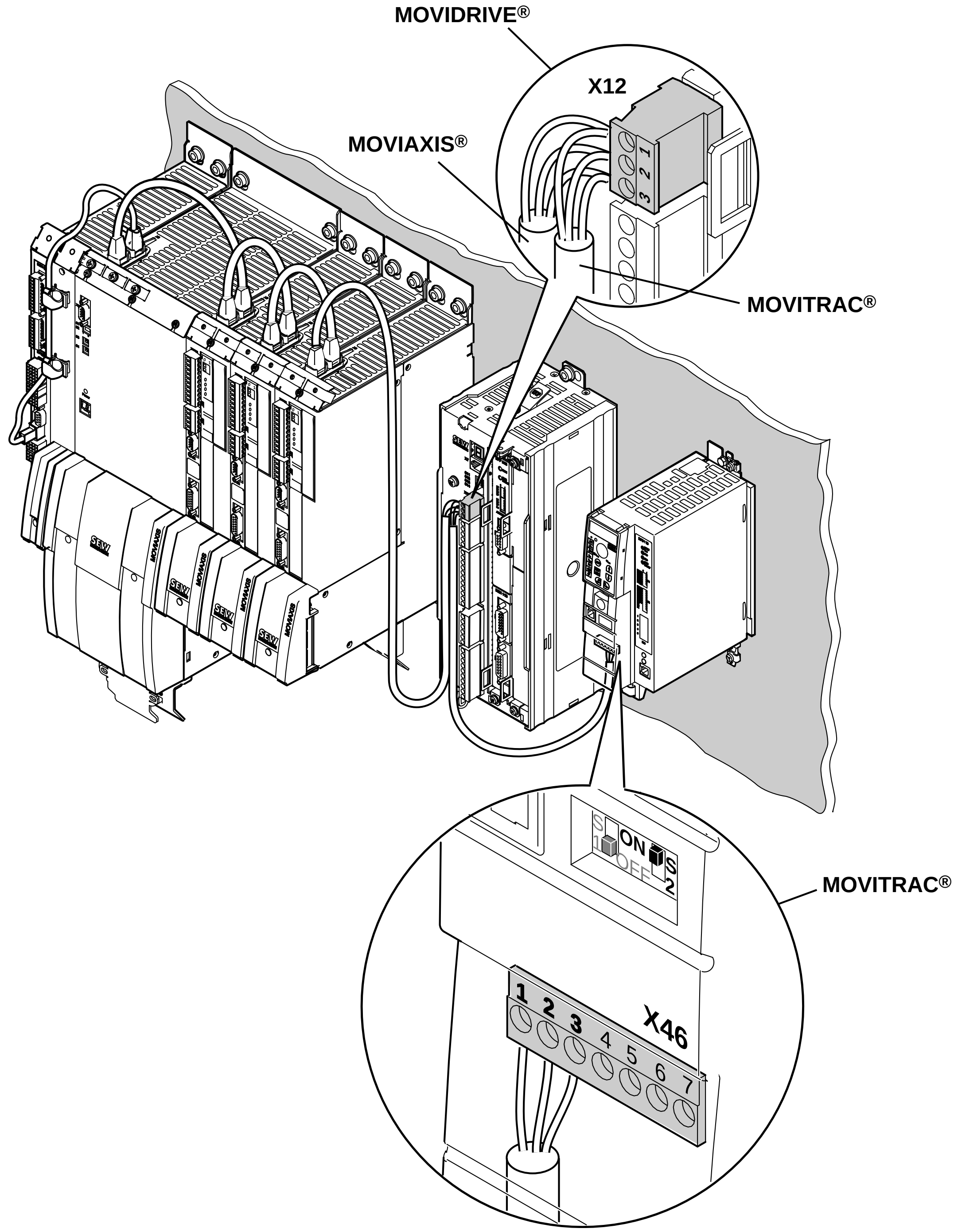
64784AXX

[1] Terminating resistor

Overview of system connection cables

Type	Part number	Description
CAN system cable	0819 692 3	System cable UFR41B gateway CAN 1 post connector (or CAN 2) to MOVIAXIS® supply/regenerative power module CAN 1 system bus RJ45, length: 750 mm
CAN1 connection cable, 750 mm, RJ45-RJ45	0819 7261	CAN1 connection cable between MOVIAXIS® axis system and MOVIAXIS® axis system, length: 750 mm
CAN1 connection cable, 3000 mm, RJ45-RJ45	0819 8993	CAN1 connection cable between MOVIAXIS® axis system and MOVIAXIS® axis system, length: 3000 mm
CAN2 adapter cable	1810 1607	CAN2 post connector between master module and CAN2 SUB-D9 MOVIAXIS®, length: 500 mm
CAN2 connection cable	1810 1585	CAN2 SUB-D9 MOVIAXIS® and CAN2 SUB-D9 MOVIAXIS®, to connect 3 axis modules
CAN2 connection cable	1810 1593	CAN2 SUB-D9 MOVIAXIS® and CAN2 SUB-D9 MOVIAXIS®, to connect 4 axis modules
Terminating resistor CAN 2	1810 1615	Terminating resistor for CAN 2 connections between axis modules

Wiring diagram for MOVIAXIS®, MOVIDRIVE® B and MOVITRAC® B on CAN 1 system bus



64783AXX

Overview of system connection cables

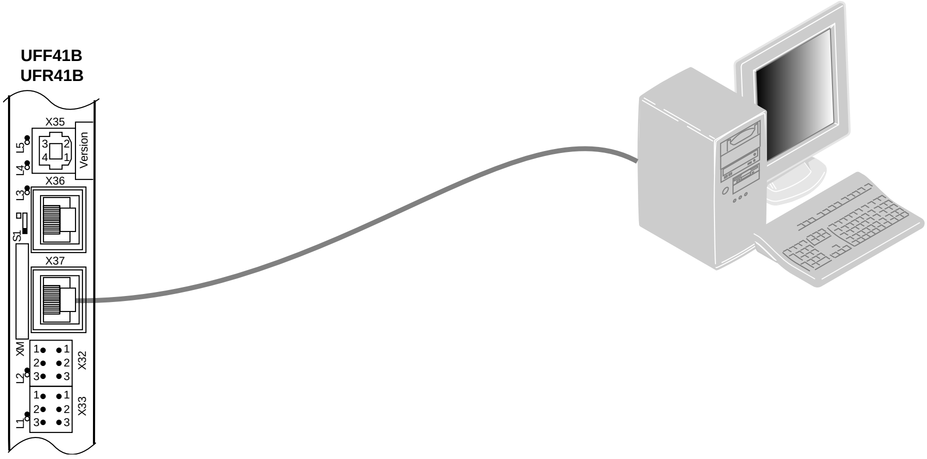
Type	Part number	Description
CAN1 connection cable, 750 mm, RJ45 litz wire	0819 7288	CAN connection cable MOVIAXIS® axis system to MOVIDRIVE® and MOVITRAC®, length: 750 mm
CAN1 connection cable, 3000 mm, RJ45 litz wire	0819 7563	CAN connection cable MOVIAXIS® axis system to MOVIDRIVE® and MOVITRAC®, length: 3000 mm

4.3.3 Connecting SBUS^{plus} system bus (terminal X36)

Terminal X36 is intended for connecting a system bus based on EtherCAT (SBUS^{plus}).

4.3.4 Ethernet interface terminal (terminal X37)

You can connect an engineering PC to the Ethernet interface (terminal X37).



65057AXX

The Ethernet interface (X37) supports auto crossing auto negotiation for baud rate and duplex mode. The IP parameters are defined depending on DIP switch S1 (see section "DIP switches S1 default IP address").

In addition to the engineering access via terminal X37, there is another engineering access via PROFIBUS (see section "Operation of MOVITOOLS® MotionStudio").

4.4 Status LED of the UFR41B fieldbus gateway

LED L1 (CAN 1 status)

The LED **L1** indicates the **status** of the **CAN 1** system bus.

Status of the L1 LED	Diagnostics	Remedy
Orange	The CAN 1 system bus is being initialized.	-
Green	The CAN 1 system bus is initialized.	
Flashing green (0.5 Hz)	The CAN 1 system bus is currently in SCOM suspend mode.	
Flashing green (1 Hz)	The CAN 1 system bus is currently in SCOM On mode.	
Red	The CAN 1 system bus is off (BUS-OFF).	1. Check and correct the cabling of the CAN 1 system bus. 2. Check and correct the baud rate set for the CAN 1 system bus. 3. Check and correct the terminating resistors of the CAN 1 system bus.
Flashing red (1 Hz)	Warning on the CAN 1 system bus.	1. Check and correct the cabling of the CAN 1 system bus. 2. Check and correct the baud rate set for the CAN 1 system bus.

LED L2 (CAN 2 status)

The LED **L2** indicates the **status** of the **CAN 2** system bus.

Status of the L2 LED	Diagnostics	Remedy
Orange	The CAN 2 system bus is being initialized.	-
Green	The CAN 2 system bus is initialized.	-
Flashing green (0.5 Hz)	The CAN 2 system bus is currently in SCOM suspend mode.	-
Flashing green (1 Hz)	The CAN 2 system bus is currently in SCOM On mode.	-
Red	The CAN 2 system bus is off (BUS-OFF).	1. Check and correct the cabling of the CAN 2 system bus. 2. Check and correct the baud rate set for the CAN 2 system bus. 3. Check and correct the terminating resistors of the CAN 2 system bus.
Flashing red (1 Hz)	Warning on the CAN 2 system bus.	1. Check and correct the cabling of the CAN 2 system bus. 2. Check and correct the baud rate set for the CAN 2 system bus.

LED L3 (program status)

LED **L3** indicates the **status** of the **gateway program**.

Status of L3	Diagnostics	Remedy
Green	Gateway program is running.	-
Off	No gateway program is loaded.	Load a gateway program into the controller.
Flashing orange (1 Hz)	Program has stopped.	Bootloader update required (see section "SD memory card type OMG4.B")

LED 4 (PLC status)

LED **L4** indicates the **firmware status** of the fieldbus gateway.

Status of the L4 LED	Diagnostics	Remedy
Flashing green (1 Hz)	The firmware of the fieldbus gateway is running properly.	-
Red	• No SD card plugged in. • File system of the SD card corrupt.	
Flashing orange (1 Hz)	Program has stopped.	Bootloader update required (see section "SD memory card type OMG4.B")

LED L5 (user)

LED **L5** is lit up red if the gateway program has detected an error and if this error can only be eliminated after diagnostics with MOVITOOLS® MotionStudio.

4.5 DIP switch S1 default IP address

With DIP switch S1, you can set a default IP address for the Ethernet connection (X37). The set IP address is applied in the next boot process.

S1 switch setting	Meaning
Top	IP parameter: • IP address: 192.168.10.4 • Subnet mask: 255.255.255.0 • Standard gateway: 1.0.0.0
Bottom	The IP parameters defined on the memory card of the UFR41B gateway are used. The IP parameters for engineering interface X37 are entered in the file '...\System\NetConfig.cfg' in section 'Ethernet 2'. You can adjust the file using a text editor (e.g. Notepad).

4.6 SD memory card type OMG4.B

The SD memory card type OMG4.B is required for operating the UFR41B fieldbus gateway and contains the firmware, the gateway program, and the gateway configuration. With a MOVIAXIS® axis module, it is also used for data backup and automatic parameterization in case an axis needs to be replaced.

The SD memory card type OMG4.B is included in the scope of delivery of the UFR41B fieldbus gateway.

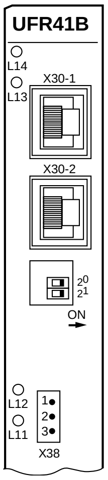
Only use type OMG4.B memory cards in a UFR41B fieldbus gateway.

Bootloader update

When the LEDs L3 and L4 flash orange at a 1 Hz frequency after power-on, a bootloader update is required. Proceed as follows:

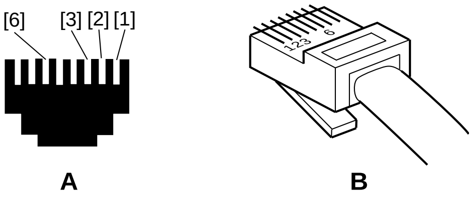
- Do not switch off the power supply during the entire process.
- Press the reset button T1 on the front of the UFR41B fieldbus gateway for 3 seconds. When the bootloader update starts, only LED 4 is flashing.
- The bootloader update has been successful when L4 flashes green.

4.7 Connecting the UFR41B fieldbus gateway to an Ethernet network

Front view UFR41B fieldbus gateway	Designation	LED DIP switch Terminal	Function
 UFR41B L14 L13 X30-1 X30-2 2⁰ 2¹ ON L12 L11 1 2 3 X38 65052AXX	**LED**	 **L14** **L13** **L14** **L13** **L12** **L11**	In EtherNet/IP and Modbus/TCP operation: MODULE STATUS NETWORK STATUS In PROFINET operation: RUN BUS FAULT Reserved Reserved
	X30-1: Ethernet connection **LED Link (green)** **LED Activity (yellow)**		
	X30-2: Ethernet connection **LED Link (green)** **LED Activity (yellow)**		
	DIP switch	**2^0 = ON**	Resets the address parameters to their default values and deactivates DHCP • IP address: 192.168.10.4 • Subnet mask: 255.255.255.0 • Gateway: 192.168.10.4
		2^1 = ON **2^1 = OFF**	EtherNet/IP and Modbus/TCP protocol is active PROFINET protocol is active
	X38: CAN for safety-relevant communication	**X38:1** **X38:2** **X38:3**	Reserved Reserved Reserved

4.8 Pin assignment X30-1, X30-2 and X37

Use prefabricated, shielded RJ45 plug connectors compliant with IEC 11801 edition 2.0, category 5.



54174AXX

A	View from front	B	View from back
[1]	Pin 1 TX+ Transmit Plus	[2]	Pin 2 TX- Transmit Minus
[3]	Pin 3 RX+ Receive Plus	[6]	Pin 6 RX- Receive Minus

Connecting UFR41B fieldbus gateway to Ethernet

To connect UFR41B to the Ethernet, connect the Ethernet interface (X30-1, X30-2 or X37) to the other network stations using a category 5, class D shielded twisted-pair cable in accordance with IEC 11801 edition 2.0. The integrated switch provides support for implementing a line topology using X30-1 and X30-2, and offers auto crossing functions.

TIPS

- According to IEC 802.3, the maximum cable length for 10/100 MBaud Ethernet (10BaseT / 100BaseT), e.g. between two network stations, is 100 m.
- We recommend that you do not directly connect non-SEW end devices to the UFR41B option in order to minimize the load on the end devices in EtherNet/IP networks caused by undesired multicast data traffic. Connect non-SEW devices via a network component that supports the IGMP snooping functionality (e.g. managed switch).

  Managed switches with IGMP snooping functionality is not required for PROFINET IO and Modbus TCP networks.

4.9 Shielding and routing bus cables

Only use shielded cables and connection elements that meet the requirements of category 5, class D according to IEC 11801 edition 2.0.

Correct shielding of the bus cable attenuates electrical interference that can occur in industrial environments. The following measures ensure the best possible shielding:

- Manually tighten the mounting screws on the connectors, modules, and equipotential bonding conductors.
- Use only connectors with a metal housing or a metallized housing.
- Connect the shielding in the connector over a wide surface area.
- Apply the shielding of the bus cable on both ends.
- Route signal and bus cables in separate cable ducts. Do not route them parallel to power cables (motor leads).
- Use metallic, grounded cable racks in industrial environments.
- Route the signal cable and the corresponding equipotential bonding close to each other using the shortest possible route.
- Avoid using plug connectors to extend bus cables.
- Route the bus cables closely along existing grounding surfaces.

CAUTION

In case of fluctuations in the ground potential, a compensating current may flow via the bilaterally connected shield that is also connected to the protective earth (PE). Make sure you supply adequate equipotential bonding according in accordance with relevant VDE regulations in such a case.

4.10 The integrated Ethernet switch

You can use the integrated Ethernet switch to implement line topologies using X30-1 and X30-2 known from the fieldbus technology. Other bus topologies, such as star or tree, are also possible. Ring topologies are not supported.

TIP

The number of Industrial Ethernet switches connected in line impacts on the telegram runtime. If a telegram passes through the units, the telegram runtime is delayed by the store & forward function of the Ethernet switch:

- for a telegram length of 64 bytes by approximately 10 µs (at 100 Mbit/s)
- for a telegram length of 1500 bytes by approximately 130 µs (at 100 Mbit/s)

This means that the more units a telegram has to pass through, the higher the telegram runtime is.

Auto crossing

The two ports leading out of the Ethernet switch have auto crossing functionality. This means that they can use both patch and cross-over cables to connect to the next Ethernet station.

Auto negotiation

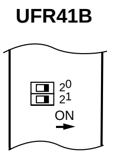
The baud rate and the duplex mode is negotiated by both Ethernet nodes when establishing the connection. For this purpose, both Ethernet ports of the EtherNet/IP connection support an auto negotiation functionality and work with a baud rate of either 100 Mbit or 10 Mbit in full duplex or half-duplex mode.

Notes on multicast handling

- The integrated Ethernet switch does not provide a filter function for Ethernet multicast telegrams. Multicast telegrams that are usually sent in Ethernet/IP networks from the adapters to the scanners (PLC) are passed on to all switch ports.
- IGMP snooping (managed switch) is not supported.
- SEW-EURODRIVE therefore recommends to connect the UFR41B option in EtherNet/IP networks only with network components that support IGMP snooping (e.g. managed switch) or that have safety mechanisms integrated against excess multicast load (e.g. units from SEW-EURODRIVE). Units that do not have this integrated function can fail due to high network loads. This restriction does not apply to PROFINET IO or MODBUS TCP networks.

4.11 Setting the DIP switches

	TIP
ℹ	De-energize the UFR41B fieldbus gateway before you change a DIP switch setting. The DIP switch settings are adopted during initialization only.



65053AXX

2^0 (Def IP)

if the switch "2^0" is set to "1" (= right = ON), the following default IP address parameters are set when the DC 24 V backup voltage is switched on:

- IP address: 192.168.10.4
- Subnet mask: 255.255.255.0
- Default gateway: 192.168.10.4
- P785 DHCP / Startup configuration: Saved IP parameters (DHCP is deactivated)

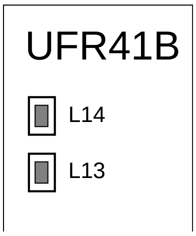
2^1 (protocol)

DIP switch "2^1" is used to set the protocol that is used for communication.

- 2^1 = "1" (= right = ON): The EtherNet/IP and Modbus TCP/IP fieldbus protocol is active.
- 2^1 = "0" (= left = OFF): The PROFINET fieldbus protocol is active.

4.12 Status LED of the UFR41B fieldbus gateway

The LEDs of the UFR41B fieldbus gateway indicate the current status of the UFR41B option and the fieldbus system. The LEDs have a different meaning depending on the set protocol.



65054AXX

4.12.1 Status LED in EtherNet/IP and Modbus/TCP operation

Chapter 9 provides a summary of the status of the fieldbus interface corresponding to the LED status.

LED L13 (NETWORK STATUS)

The LED **L13 (NETWORK STATUS)** indicates the state of the fieldbus system.

States of the NETWORK STATUS LED	Meaning
Off	The UFR41B option does not yet have any IP parameters.
Flashing green/red	The UFR41B option card performs an LED test.
Flashing green	There is no controlling IO connection.
Green	There is a controlling EtherNet/IP or Modbus/TCP connection.
Red	Conflict detected in the assigned IP addresses. Another station in the network uses the same IP address.
Flashing red	The previously established controlling IO connection is in timeout status. The status is reset by restarting communication.

LED L14 (MODULE STATUS)

LED **L14 (MODULE STATUS)** indicates that the bus electronics are operating correctly.

States of the MODULE STATUS LED	Meaning
Off	The UFR41B option card is either not supplied with voltage or it is faulty.
Flashing green	• If the NETWORK STATUS LED is off at the same time, the TCP/IP stack of the UFR41B option card will be started. If this status continues and DHCP is activated, the UFR41B option card waits for data from the DHCP server. • If the NETWORK STATUS LED is flashing green at the same time, the application of the UFR41B option card is started.
Flashing green/red	The UFR41B option card performs an LED test.
Green	The UFR41B option card is in normal operating state.
Red	The UFR41B option card is in fault state.
Flashing red	Conflict detected in the assigned IP addresses. Another station in the network uses the same IP address.

4.12.2 Status LED in PROFINET operation

L13 LED (BUS FAULT)

The LED **L13 (BUS FAULT)** indicates the status of the PROFINET.

Status of the L13 LED	Cause of error	Remedy
Off	• PROFINET IO device is currently exchanging data with the PROFINET IO controller (data exchange).	-
Flashing green **Flashing green/red**	• The flashing function in the PROFINET IO controller configuration is activated to visually localize the stations.	-
Red	• Connection to the PROFINET IO controller has failed. • PROFINET IO device does not detect a link • Bus interruption • PROFINET IO controller is not in operation	• Check the PROFINET connection of the UFR41B fieldbus gateway • Check the PROFINET IO controller • Check the cabling of your PROFINET network
Yellow **Flashing yellow**	• The STEP 7 hardware configuration contains a module that is not permitted.	• Switch the STEP 7 hardware configuration to ONLINE and analyze the status of the components of the slots in the PROFINET IO device.

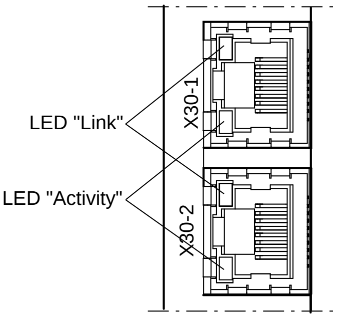
LED L14 (RUN)

LED **L14 (RUN)** indicates that the bus electronics are operating correctly.

Status of the L14 LED	Cause of error	Remedy
Green	• UFR41B hardware OK. • Functions properly	-
Off	• UFR41B is not ready for operation.	• Switch the unit on again. Consult SEW service if the error occurs again.
Red	• Error in the UFR41B hardware	
Flashing green	• Hardware of the UFR41B does not boot up.	
Flashing yellow		• Switch the unit on again. Set default IP address parameter using DIP switch "S1". Consult SEW service if the error occurs again.
Yellow		• Switch the unit on again. Consult SEW service if the error occurs again.

4.12.3 Link / Activity LEDs

The two LEDs **Link (green)** and **Activity (yellow)**, integrated in the RJ45 plug connectors (X30-1, X30-2), indicate the status of the Ethernet connection.



63365AXX

LED/status	Meaning
Link/green	There is an Ethernet connection.
Link/off	There is no Ethernet connection.
Link/flashes	Function for localizing in the SEW address editor (see chapter "Operating MOVITOOLS® MotionStudio")
Activity/ yellow	Data is currently being exchanged via Ethernet.

4.13 TCP / IP addressing and subnetworks

Introduction

The settings for the address of the IP protocol are made using the following parameters:

- MAC address
- IP address
- Subnetwork mask
- Standard gateway

The addressing mechanisms and subdivision of the IP networks into sub-networks are explained in this chapter to help you set the parameters correctly.

MAC address

The MAC address (Media Access Controller) is the basis for all address settings. The MAC address is a worldwide unique 6-byte value (48 bits) assigned to the Ethernet device. SEW Ethernet devices have the MAC address 00-0F-69-xx-xx-xx. The MAC address is difficult to handle for larger networks. This is why freely assignable IP addresses are used.

IP address

The IP address is a 32 bit value that uniquely identifies a station in the network. An IP address is represented by four decimal numbers separated by decimal points.

Example: 192.168.10.4

Each decimal number stands for one byte (= 8 bits) of the address and can also be represented using binary code (see following table).

Byte 1		Byte 2		Byte 3		Byte 4
11000000	.	10101000	.	00001010	.	00000100

The IP address comprises a network address and a station address (see following table).

Network address	Station address
192.168.10	4

The part of the IP address that denotes the network and the part that identifies the station is determined by the network class and the subnetwork mask.

Station addresses cannot consist of only zeros or ones (binary) because they represent the network itself or a broadcast address.

Network classes

The first byte of the IP address determines the network class and as such represents the division into network addresses and station addresses.

Value range Byte 1	Network class	Complete network address (Example)	Meaning
0 ... 127	A	10.1.22.3	10 = Network address 1.22.3 = Station address
128 ... 191	B	172.16.52.4	172.16 = Network address 52.4 = Station address
192 ... 223	C	192.168.10.4	192.168.10 = Network address 4 = Station address

This rough division is not sufficient for a number of networks. They also use an explicit, adjustable subnetwork mask.

Subnetwork mask

A subnetwork mask is used to divide the network classes into even finer sections. Like the IP address, the subnetwork mask is represented by four decimal numbers separated by decimal points.

Example: 255.255.255.128

Each decimal number stands for one byte (= 8 bits) of the subnetwork mask and can also be represented using binary code (see following table).

Byte 1		**Byte 2**		**Byte 3**		**Byte 4**
11111111	.	11111111	.	11111111	.	10000000

If you compare the IP addresses with the subnetwork masks, you see that in the binary representation of the subnetwork mask all ones determine the network address and all the zeros determine the station address (see following table).

		Byte 1		**Byte 2**		**Byte 3**		**Byte 4**
IP address	decimal	192	.	168.	.	10	.	129
	binary	11000000	.	10101000	.	00001010	.	10000001
Subnetwork mask	decimal	255	.	255	.	255	.	128
	binary	11111111	.	11111111	.	11111111	.	10000000

The class C network with the address 192.168.10. is further subdivided into 255.255.255.128 using the subnetwork mask. Two networks are created with the address 192.168.10.0 and 192.168.10.128.

The following station addresses are permitted in the two networks:

- 192.168.10.1 ... 192.168.10.126
- 192.168.10.129 ... 192.168.10.254

The network stations use a logical AND operation for the IP address and the subnetwork mask to determine whether there is a communication partner in the same network or in a different network. If the communication partner is in a different network, the standard gateway is addressed for passing on the data.

Standard gateway

The standard gateway is also addressed via a 32-bit address. The 32-bit address is represented by four decimal numbers separated by decimal points.

Example: 192.168.10.1

The standard gateway establishes a connection to other networks. In this way, a network station that wants to address another station can use a logical AND operation with the IP address and the subnetwork mask to determine whether the desired station is located in the same network. If this is not the case, the station addresses the standard gateway (router), which must be part of the actual network. The standard gateway then takes on the job of transmitting the data packages.

If for the standard gateway, the same address is set as for the IP address, the standard gateway is deactivated. The address of the standard gateway and the IP address must be in the same subnetwork.

DHCP (Dynamic Host Configuration Protocol)

Instead of setting the three parameters IP address, subnetwork mask and standard gateway manually, they can be assigned in an automated manner by a DHCP server in the Ethernet network.

This means the IP address is assigned from a table, which contains the allocation of MAC address to IP address.

Parameter P785 indicates whether the UFR41B option expects the IP parameters to be assigned manually or via DHCP.

4.14 Setting the IP address parameters

Initial startup

If the EtherNet/IP and MODBUS TCP protocol is set using DIP switch, the default protocol for the UFR41B option will be "DHCP" (**D**ynamic **H**ost **C**onfiguration **P**rotocol). This means the option card expects its IP address parameters from a DHCP server.

TIP

Rockwell Automation provides a DHCP server free-of-charge on their homepage. The tool is known as "BOOTP Utility" and can be downloaded from the Allen Bradley website.

Once the DHCP server has been configured and the settings have been made for the subnetwork screen and the standard gateway, the UFR41B must be inserted in the assignment list of the DHCP server. The MAC ID of the UFR41B option is allocated a valid IP address.

TIP

The configured IP address parameters are adopted permanently by the parameter set when DHCP is deactivated after the IP address has been assigned.

Changing the IP address parameters after successful initial startup

TIP

If the PROFINET IO setting is avtive via DIP switch, then the IP address will be assigned using the engineering system of the IO controller (see chapter "PROFINET IO configuration").

If the UFR41B was started using a valid IP address, you can also access the IP address parameters via the Ethernet interface.

There are various ways to change the IP address parameters via Ethernet:

- Using the MOVITOOLS® MotionStudio software
- Using the EtherNet/IP TCP/IP interface object (see section "EtherNet/IP CIP object directory")
- Using the SEW Address Editor

You can also change the IP address parameters via the other interface of UFR41B.

If the IP address parameters are assigned to the UFR41B option via DHCP server, then you can only change the parameters by adjusting the settings of the DHCP server.

The options listed above for changing the IP address parameters only come into effect once the supply voltages (DC 24 V) have been switched off and back on again.

TIP

If PROFINET IO is used, the IP address must be changed in the engineering system of the IO controller because the IO controller overwrites the IP address of the fieldbus gateway after a restart (power off and on).

Deactivating / activating DHCP

The type of IP address allocation is determined by the setting of the attribute *Configuration Control* of the EtherNet/IP TCP / IP interface object. The value is displayed or modified in the parameter *P785 DHCP / Startup Configuration*.

- Setting "Saved IP parameters"

  The saved IP address parameters are used.

- Setting "DHCP"

  The IP address parameters are requested by a DHCP server.

  If you use the DHCP server from Rockwell Automation, you can activate or deactivate the DHCP via a button. In this case, an EtherNet/IP telegram is sent to the TCP/IP interface object of the station that is being addressed.

Resetting the IP address parameters

If you do not know the IP address parameters and there is no other interface for reading the IP address, you can reset the IP address parameters to the default values using DIP switch "2^0".

This action resets the UFR41B option to the following default values:

- IP address: 192.168.10.4
- Subnetwork mask: 255.255.255.0
- Default gateway: 192.168.10.4
- DHCP / Startup Configuration: Saved IP parameters (DHCP is deactivated)

Proceed as follows to reset the IP address parameters to the default values:

- Switch off the 24 V DC supply voltage and the mains voltage.
- Set the DIP switch "2^0" on the UFR41B option to "1".
- Switch the DC 24 V supply voltage and the line voltage back on.

SEW Address Editor

You can also use the SEW Address Editor to access the IP settings of UFR41B without the Ethernet settings of the PC and UFR41B having to match.

The IP settings of all SEW units can be made and displayed in the local subnetwork using Address Editor in MOVITOOLS® MotionStudio (see chapter 10).

- In this way, you can determine the PC settings required to provide for an access with the required diagnostics and engineering tools via Ethernet while the installation is in progress.
- When starting up a unit, the IP settings for the UFR41B can be assigned without changing the network connections or PC settings.

TIP

- DHCP remains deactivated when you reset the DIP switch "2^0" (Def IP) to "0". You can re-activate DHCP via the EtherNet/IP TCP/IP interface object (see section 'EtherNet/IP CIP object directory'), via the parameter, or via the DHCP server from Rockwell Automation.
- DHCP is activated again when the values are reset to the factory setting.

4.15 Procedure for replacing the unit

- Use the OMG4.B memory card of the old unit when replacing the fieldbus gateway. Simply insert the memory card in the new unit. All configuration data are stored on the memory card, including the PROFINET device name.
- If DIP switch "2^0" (Def IP) is set to "1" (= ON) at the UFR41B option, then DIP switch "2^0" (Def IP) of the new UFR41B must also be set to "1" (= ON). Other IP parameter settings are not required.
- If DHCP is active, the assignment list of the DHCP server must be updated when the UFR41B option is replaced. The MAC address of the UFR41B option is printed on its front panel for this purpose.
- If DHCP is not active, the IP parameters saved on the memory card of UFR41B will be used.

  If the memory card of UFR41B is not plugged into the new unit when replacing the old one, you will have to perform a complete startup of the new UFR41B (if DHCP is not active including the IP parameters). Instead, you can load a data backup created with the MOVITOOLS® MotionStudio software to the new unit.

5 Configuring the UFx41B Fieldbus Gateway and Inverter

5.1 Description of the gateway functions

5.1.1 Introduction

With the UFF41B and UFR41B fieldbus gateways, SEW-EURODRIVE offers innovative solutions for integrating SEW inverter technology in fieldbus systems.

For this purpose, process data of the higher-level control in the fieldbus gateway are processed and sent via CAN (SBus) to the devices connected to the fieldbus gateway. Type UFx41B fieldbus gateways can transmit up to 64 process data (PD) from the fieldbus to up to 16 lower-level slave units. The data length per slave unit is limited to 16 process data.

Two different unit configurations are supported:

- Auto setup configuration

  For automatic configuration of the fieldbus gateway and connected devices.
- Customized configuration

  For individual configuration of the process data length and the CAN connection of the individual slave units.

Special features of the UFx41B fieldbus gateways are data backup and data restoration (see chapter "Data Backup", section "Restore mechanism") after replacement of a slave unit. For this purpose, all parameters of the connected slave units are saved on the SD card of the fieldbus gateway and a possible unit replacement is monitored. When a unit is replaced, the fieldbus gateway automatically loads the unit parameters to the replaced unit.

The fieldbus gateway is configured in MOVITOOLS® MotionStudio using the "UFx Gateway Configurator" tool.

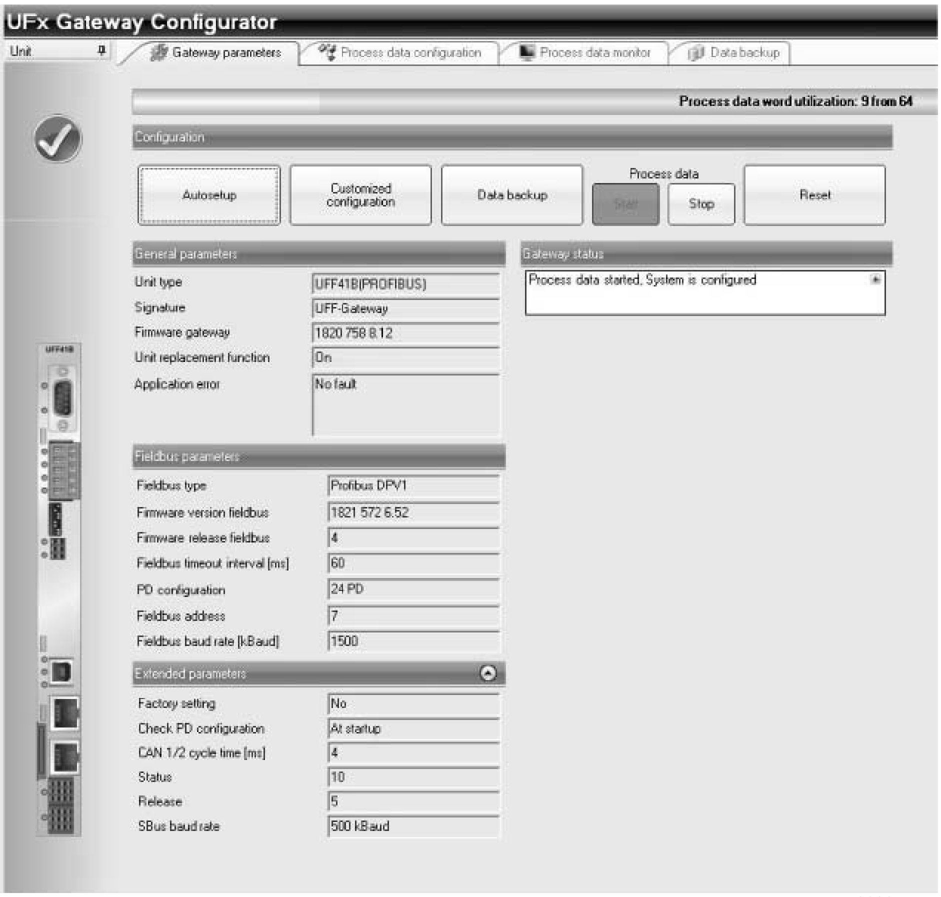
5.1.2 Autosetup

The "Autosetup" function is activated in the "UFx Gateway Configurator" tool. Autosetup results in automatic configuration of the fieldbus gateway and the slave units connected to it, which optimally cover a wide range of applications.

The "autosetup" function performs the following configurations automatically:

- Stopping process data communication in direction of the SBus
- Scanning the CAN 1 system bus to detect the connected units (MOVIAXIS®, MOVIDRIVE® B and MOVITRAC® B; up to max. 16 units)
- Assigning the process data length: 6 process data with MOVIAXIS® and 3 process data with MOVIDRIVE® B and MOVITRAC® B
- Configuring the necessary process data objects (PDO) of the MOVIAXIS® axis modules
- Saving the configuration in the UFx41B fieldbus gateway (no data backup)

- Starting process data communication



Process data word utilization: 9 from 64

Configuration

General parameters

Unit type	UFF41B(PROFIBUS)
Signature	UFF-Gateway
Firmware gateway	1820 758 8.12
Unit replacement function	On
Application error	No fault

Gateway status

Process data started, System is configured

Fieldbus parameters

Fieldbus type	Profibus DPV1
Firmware version fieldbus	1821 572 6.52
Firmware release fieldbus	4
Fieldbus timeout interval [ms]	60
PD configuration	24 PD
Fieldbus address	7
Fieldbus baud rate [kBaud]	1500

Extended parameters

Factory setting	No
Check PD configuration	At startup
CAN 1/2 cycle time [ms]	4
Status	10
Release	5
SBus baud rate	500 kBaud

12082AEN

During unit scan, the first 16 units found in the slave unit configuration saved in the fieldbus gateway will apply.

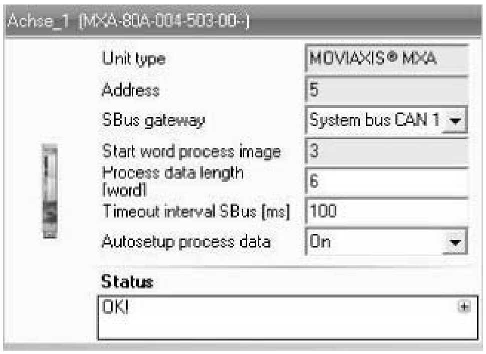
If the value of 64 PD is exceeded due to the process data lengths set for the individual slave units, the gateway application will automatically reduce the process data length of the slave units. In this case, 3 PD are set for MOVIDRIVE® B and MOVITRAC® B slave units. The remaining free PD length will be divided by the number of MOVIAXIS® slave units. This is the resulting process data length for the individual MOVIAXIS® units. The procedure applies no matter whether the autosetup function is enabled or not.

	TIP
	"Autosetup" assumes that all slave units are connected to the CAN 1 system bus. Scanning is performed using the CAN 1 system bus only. The start words in the process image are set in such a way that the data of the slaves follows one another without overlapping. The auto setup configuration is saved in the UFx41B fieldbus gateway and is checked by scanning the slave units each time power supply is switched on. To ensure successful communication and configuration of MOVIAXIS® units, the MOVIAXIS® parameter setting level must be set to "Planning Engineer".

5.1.3 Customized configuration

The "customized configuration" function allows for configuring the process data length individually and for using the CAN 2 terminal on the fieldbus gateway. The CAN cycle time can be reduced by dividing the slave units among the two CAN interfaces of the fieldbus gateway. The data transmission performance can be increased in this way.

Customized configuration means that users can configure the process data length for each slave unit, the start word in the process image in direction of the fieldbus, and the SBus (CAN 1 or CAN 2). Status word and data length are the same both for the process input and process output data of the slave unit.



Achse_1 (MXA-80A-004-503-00-)

Unit type	MOVIAXIS® MXA
Address	5
SBus gateway	System bus CAN 1
Start word process image	3
Process data length [word]	6
Timeout interval SBus [ms]	100
Autosetup process data	On

Status

OK!

12103AEN

The fieldbus gateway uses these data to automatically determine the cycle time for the CAN interfaces as well as the number, data length, and CAN-IDs of process data objects (PDO) on the SBus. The duration of the cycle time is always the same for both CAN interfaces.

Pressing the [Apply configuration] button saves the configuration data in the fieldbus gateway. These are the number of slave units, their process data length, their connection to the CAN1 or CAN2 system bus, and their timeout interval. Additionally, the settings required for establishing the communication with the fieldbus gateway are made automatically in the MOVIAXIS® slave units. For MOVIAXIS® units with disabled "autosetup of process data" function ("autosetup process data" selection field "off"), the user has to set the parameters for the process data in the slave units accordingly.

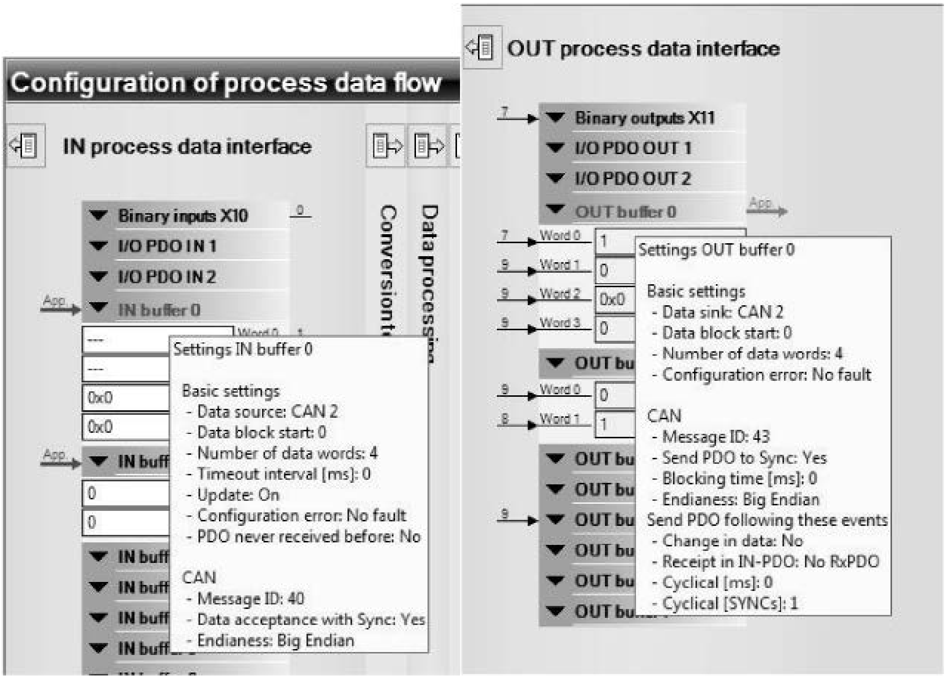
Changes made to the process data configuration in the fieldbus gateway will take effect in the fieldbus gateway by pressing the [Apply configuration] button.

5.1.4 Configuring fieldbus gateway and slave units

If the "autosetup" or "customized configuration" functions are performed using the UFx Gateway Configurator, then the slave unit parameters (MOVIAXIS®, MOVIDRIVE® B and MOVITRAC® B) described in the following sections have to be made.

Setting the MOVIAXIS® servo inverter

Process data communication is automatically configured in the axis module for each MOVIAXIS® slave unit if the fieldbus gateway was configured using the "autosetup" function, or, in the case of "customized configuration" of this slave unit, if the the process data objects required for communication between fieldbus gateway and MOVIAXIS® axis module are configured. The unit-internal further processing of process data depends on the application and is not affected by the configuration by the fieldbus gateway.



Settings IN buffer 0

Basic settings
- Data source: CAN 2
- Data block start: 0
- Number of data words: 4
- Timeout interval [ms]: 0
- Update: On
- Configuration error: No fault
- PDO never received before: No

CAN
- Message ID: 40
- Data acceptance with Sync: Yes
- Endianess: Big Endian

Settings OUT buffer 0

Basic settings
- Data sink: CAN 2
- Data block start: 0
- Number of data words: 4
- Configuration error: No fault

CAN
- Message ID: 43
- Send PDO to Sync: Yes
- Blocking time [ms]: 0
- Endianess: Big Endian

Send PDO following these events
- Change in data: No
- Receipt in IN-PDO: No RxPDO
- Cyclical [ms]: 0
- Cyclical [SYNCs]: 1

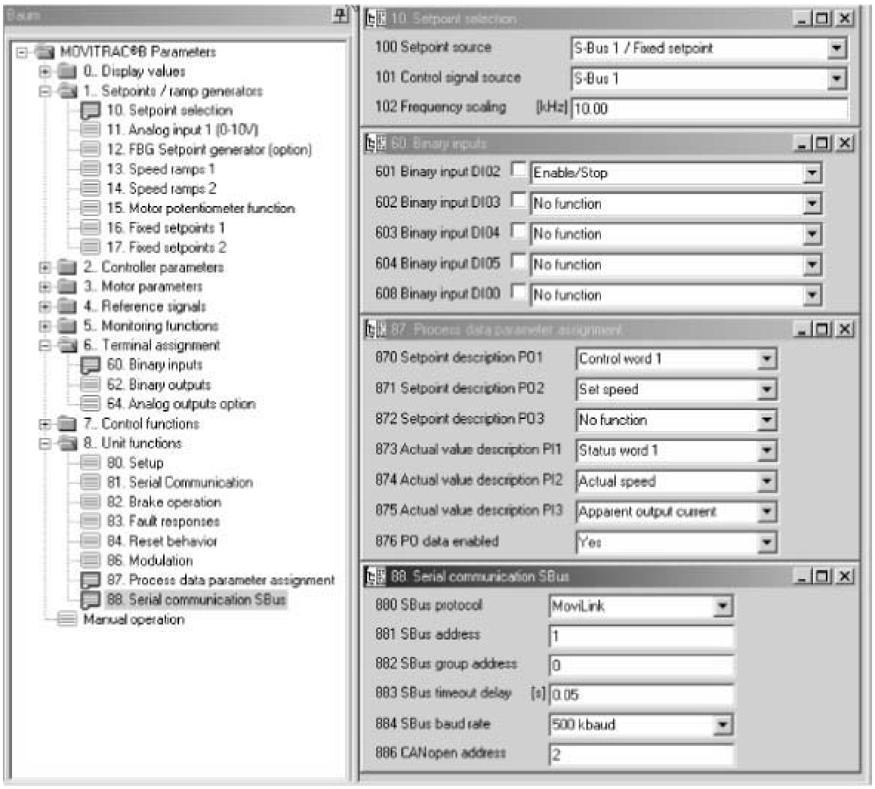
12083AEN

TIP

- It is important that no other axis-to-axis communication between the individual slave units was configured via the same CAN bus in order to ensure process data exchange and engineering between fieldbus gateway and slave units.
- If the application requires axis-to-axis communication, use the CAN2 bus of the axis module for MOVIAXIS®, and the free CAN bus for MOVIDRIVE® B.

Setting the MOVIDRIVE® B and MOVITRAC® B inverters

With MOVIDRIVE® B and MOVITRAC® B, the "autosetup" function does not automatically set the parameters. In this case, the following settings have to be made via the UFx41B fieldbus gateway for operating the MOVIDRIVE® B or MOVITRAC® B inverters (see following figure).



10. Setpoint selection

100 Setpoint source: S-Bus 1 / Fixed setpoint
101 Control signal source: S-Bus 1
102 Frequency scaling [kHz]: 10.00

60. Binary inputs

601 Binary input DI02 □ Enable/Stop
602 Binary input DI03 □ No function
603 Binary input DI04 □ No function
604 Binary input DI05 □ No function
608 Binary input DI00 □ No function

87. Process data parameter assignment

870 Setpoint description PO1: Control word 1
871 Setpoint description PO2: Set speed
872 Setpoint description PO3: No function
873 Actual value description PI1: Status word 1
874 Actual value description PI2: Actual speed
875 Actual value description PI3: Apparent output current
876 PO data enabled: Yes

88. Serial communication SBus

880 SBus protocol: MoviLink
881 SBus address: 1
882 SBus group address: 0
883 SBus timeout delay [s]: 0.05
884 SBus baud rate: 500 kbaud
886 CANopen address: 2

11845AXX

Before controlling the MOVIDRIVE® B or MOVITRAC® B inverter via the fieldbus gateway, you have to set *control signal source (P101)* and *setpoint source (P100)* to SBus1. The SBus setting1 means the inverter parameters are set for control and setpoint entry via fieldbus gateway. The inverter then responds to the process output data transmitted from the master programmable controller.

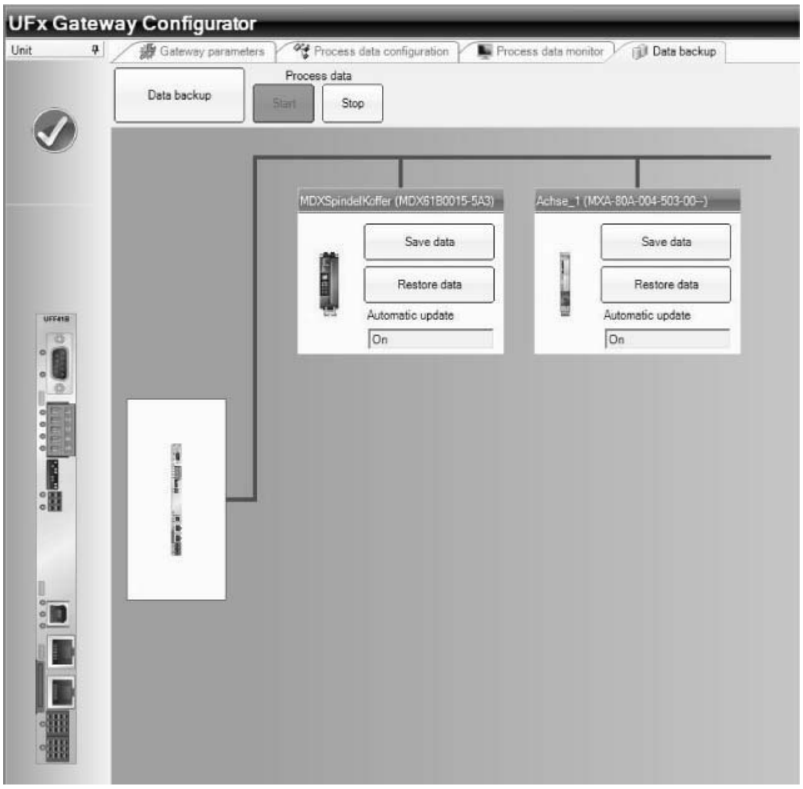
It is necessary to set the *SBus1 timeout interval (P883)* to a value other than 0 ms for the inverter to stop in the event of faulty SBus communication. We recommend a value in the range between 50 and 200 ms. Activation of the control signal source and setpoint source SBus is signaled to the higher-level controller using the "SBus mode active" bit in the status word.

Activation of the control signal source and setpoint source SBus is signaled to the machine controller using the "Fieldbus mode active" bit in the status word. For safety reasons, you must also enable the MOVIDRIVE® B inverter at the terminals for control via the fieldbus gateway. Consequently, you must wire and program the terminals in such a way that the inverter is enabled via the input terminals. The simplest way of enabling the inverter using terminals is, for example, to connect the DIØØ input terminal (function /CONTROLLER INHIBIT) for MOVIDRIVE® B, and DI01 = CW/stop for MOVITRAC® B to a +24 V signal and to program the remaining terminals to NO FUNCTION.

TIPS

- If the slave unit is connected to the CAN 1 system bus of the fieldbus gateway, set the *P881 SBus address* parameter in increasing order to values 1 - 16 . Set the basic address of the CAN 1 system bus of the axis block to values > 0 in particular when using MOVIAXIS® axis blocks.
- If the slave unit is connected to the CAN 2 system bus of the fieldbus gateway, set the *P881 SBus address* parameter in increasing order to values 17 - 34 .
- The SBus address 0 is used by the UFx41B fieldbus gateway and cannot be assigned.
- Set *P883 SBus timeout* to a value between 50 and 200 ms.
- For MOVIDRIVE® B, set *P889 / P899 Parameter channel 2* to ON

5.1.5 Data backup



MDXSpindelKoffer (MDX61B0015-5A3)
Automatic update: On

Achse_1 (MXA-80A-004-503-00--)
Automatic update: On

12090AEN

Fieldbus gateways of the type UFx41B allow for saving all parameters of the connected slave units to the SD memory card of the fieldbus gateway. Besides, the fieldbus gateway monitors a possible unit replacement and in this case loads the unit parameters automatically to the replaced unit. The parameter sets of the slave units and the configuration data of the UFx41B fieldbus gateway are centrally saved on the SD memory card of the fieldbus gateway and will be used when replacing a unit.

This means the fieldbus gateway serves as data memory for the data sets of the slave units and of the startup data of the fieldbus gateway.

Once you have taken up operation of the drive system, the data sets are copied to the data memory (SD card) when pressing the [Data backup] button. This function lets you save the parameter sets of each slave unit, their UUID (Universally Unique Identifier) and the configuration data of the fieldbus gateway itself. If the parameters of individual slave units should change after the data backup, then the change will also have to be updated in the data backup. This can be easily done by pressing the [Save data] button of the relevant slave unit.

When restarting the system, the system checks whether an axis has been replaced. If yes, the data set saved at startup will automatically be loaded into the replaced axis. This mechanism only works for units with a UUID (Universally Unique Identifier) (so far only for MOVIAXIS®).

Automatic unit update is only performed for fieldbus gateway slave units, which means for units the user has manually entered in the device list of the fieldbus gateway either using the user interface or during the system bus auto scan. Units that are connected to the SBus but are not listed in the device list of the fieldbus gateway, will neither be included in the data backup nor in the unit replacement function.

Saving data to SD memory card

The prerequisite for automatic update after a unit replacement is that the system has been taken into operation and that its data sets are available on the data memory (SD memory card for UFx) of the fieldbus gateway. These data sets are created by activating the "Data backup" function using the UFx Gateway Configurator. Make sure that the unit replacement function of the fieldbus gateway is active. To do so, set the "Unit replacement function" to "ON" on the "Gateway parameters" tab of the UFx Gateway Configurator.

Data backup means the data sets of the connected units are saved as well as their UUIDs. The configuration of the fieldbus gateway is also saved.

If you want the unit replacement function to be active for all units included in the device list, you have to enable the relevant parameters **before** activating data backup.

The user has to restart the SBus process data once data backup is completed. Bit 9 ("configured") in the gateway status indicates that the data memory contains valid data.

Restore mechanism

If the unit replacement function of the fieldbus gateway is active and bit 9 ("configured") is set, all slave units will be checked for unit replacement during startup. If a replaced unit is detected and if the axis replacement function for this slave unit is also active, the unit will be updated with the data set saved in the data memory.

If the unit replacement function for the fieldbus gateway is disabled, the units will not be checked for replacements and, consequently, the slave units will not be updated.

If an error occurs during automatic update of a slave unit, no process data communication will be established with this unit. This applies for errors occurring during the update as well as for errors while reading the UUID.

TIPS

When replacing a unit, make sure that the previous SBus address is set on the replaced units.

This is ensured when replacing a MOVIAXIS® unit if the address on the supply module is not changed and the fieldbus gateway is connected to the CAN 1 system bus of the MOVIAXIS® axis block.

With MOVIDRIVE® B and MOVITRAC® B, the addresses have to be set using parameters. This also applies to MOVIAXIS® when the gateway is connected to CAN 2 of the axis module.

Automatic unit update after a slave timeout

A possible cause for a slave timeout is a unit replacement while the system is running. The UUID of the unit is read and compared with the saved UUID as soon as the slave timeout has elapsed.

If a unit replacement is detected and the unit replacement function is activated for the fieldbus gateway and the relevant slave, and bit 9 is set in the fieldbus gateway status, then the replaced slave unit will be updated with the data set in the data memory.

The fieldbus gateway continues to send the timeout status word in the process image of the relevant slave to the fieldbus master both while the UUID is being transmitted and during a possible update of the slave unit. The process data on the SBus are not stopped. The fieldbus gateway sends "0" signals in all process data words to the relevant slave unit.

If errors occur while checking the UUID or downloading the data set, "0" are continued to be sent to the slave unit via SBus. The fieldbus gateway enters the error bit and an error code in the process image of this slave.

If timeout monitoring is disabled for a slave, no slave timeout will be signaled. This is the reason why no unit replacement verification is carried out during gateway operation. The unit replacement function during startup of the fieldbus gateway is not affected by this setting.

5.2 *Startup procedure*

5.2.1 Checking hardware installation and communication settings

- Check the CAN connection between fieldbus gateway and slave units according to the documentation.
- Check the terminating resistors (120 ohms) on the UFx41B fieldbus gateway and the last slave unit (see also chapter "Connecting the CAN 1 / CAN 2 system bus").
- Set the SBus address and baud rate (see chapter "Configuring the fieldbus gateway and slave units").

  All slave units connected to the fieldbus gateway must have different SBus addresses but the same SBus baud rate.

  You can make these settings using the keypads DBG60B, FBG11B (only for MOVITRAC® B) or using MOVITOOLS® MotionStudio (see chapter 11.7.2).

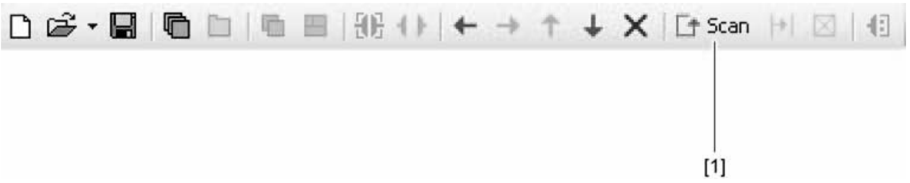
  – Set the *P881 SBus address* parameter in increasing order to values 1 - 16 if the slave unit is connected to the CAN 1 system bus of the fieldbus gateway.
  – SBus address 0 is used by the UFx41B gateway and must therefore not be used.
  – Set *P883 SBus timeout* to values between 50 to 200 ms.

5.2.2 Establishing an engineering connection

Do the following for configuring units online using MOVITOOLS® MotionStudio:

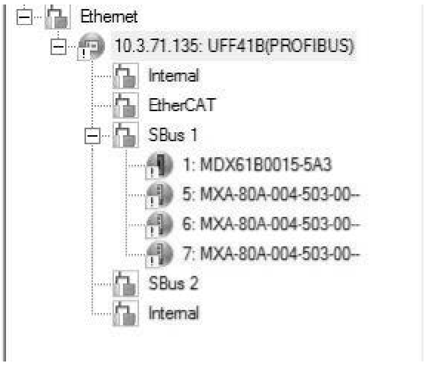
1. Start MOVITOOLS® MotionStudio from the WINDOWS® start menu using the following path:

   Start\Programs\SEW\MOVITOOLS MotionStudio
2. Create a project with name and storage location.
3. Set up communication for communicating with your units.
4. Scan the network (unit scan). To do so, click the [Start network scan] button [1] in the toolbar.



64334AXX

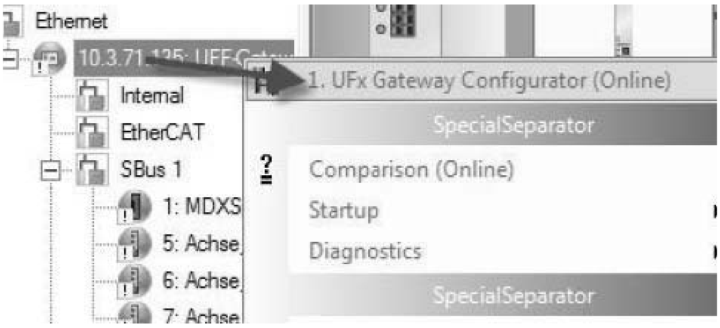
5. Make sure that all slave units connected to the fieldbus gateway are displayed after the unit scan. If no slave units are detected, check the installation (CAN bus terminating resistors). Also check whether all slave units have different SBus addresses with values higher than zero (see following figure).



Ethernet
10.3.71.135: UFF41B(PROFIBUS)
Internal
EtherCAT
SBus 1
1: MDX61B0015-5A3
5: MXA-80A-004-503-00--
6: MXA-80A-004-503-00--
7: MXA-80A-004-503-00--
SBus 2
Internal

12105AEN

6. Select the UFx41B gateway you want to configure and open the context menu with a right mouse click. You will see a number of unit-specific tools to execute various functions with the units.
7. Open the "UFx Gateway Configurator" tool (see following figure)



12104AEN

5.2.3 Configuring the fieldbus gateways

Autosetup

If you want to carry out the configuration using the "autosetup" function, press the [Autosetup] button in the UFx Gateway Configurator. All drives will be stopped.

The slave units connected to the CAN 1 system bus will be scanned and configured automatically in the case of MOVIAXIS®. The UFx Gateway Configurator displays a symbol during execution of the "Autosetup" function.

The autosetup function assigns the following process data length:

- 6 process data for MOVIAXIS®, and
- 3 process data for MOVIDRIVE® B and MOVITRAC® B

With MOVIAXIS®, all necessary process data objects (PDO) of the MOVIAXIS® axis modules are configured automatically.

With MOVIDRIVE® B and MOVITRAC® B, the SBus address, SBus timeout, and SBus baud rate have to be configured for the slave units as described in chapter "Configuring the fieldbus gateway and slave units".

The number of slave units and their settings are saved in the fieldbus gateway and are checked by scanning the slave units each time power supply is enabled.

Observe that the "autosetup" function requires that all slave units are connected to the CAN 1 system bus. Scanning is performed using the CAN 1 system bus only.

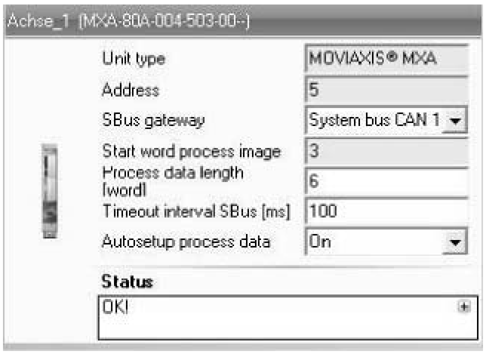
If the "Autosetup" function was executed successfully and if fieldbus communication has already been established, then the process data are started and the UFx Gateway Configurator indicates proper operation.

Customized configuration

If you want to carry out the configuration using the "customized" function, press the [Customized configuration] button in the UFx Gateway Configurator. The UFx Gateway Configurator opens the "Process data configuration" tab. Press the [Process data - Stop] button. All drives will be stopped.

The "customized configuration" functions lets you configure the process data length individually and is necessary if slave units are connected to the CAN 2 system bus of the fieldbus gateway.

The CAN cycle time can be reduced by dividing the slave units among the two CAN interfaces of the fieldbus gateway. The data transmission performance can be increased in this way.



Achse_1 (MXA-80A-004-503-00--)

Unit type	MOVIAXIS® MXA
Address	5
SBus gateway	System bus CAN 1
Start word process image	3
Process data length [word]	6
Timeout interval SBus [ms]	100
Autosetup process data	On

Status

OK!

12103AEN

Set the following for each slave unit:

- Process data length
- SBus timeout interval
- CAN interface (CAN 1 or CAN 2 system bus) to which the slave unit is connected

The entry in the "Start word process image" is determined automatically.

The start word in the process image in direction of the fieldbus as well as the process data length are the same for the process input and output data of the slave unit.

Pressing the [Apply configuration] button will perform the settings automatically in the MOVIAXIS® units where the "autosetup process data" parameter is set to "ON". If the "autosetup process data" parameter is set to "OFF", the settings in the relevant "MOVIAXIS" unit will not be made automatically so they have to be made by the user afterwards.

Pressing the [Process data - Start] button will start communication between fieldbus gateway and slave unit. The following symbol appears when communication has been established successfully.

12107AXX

5.2.4 Last settings in the slave units

Now execute the "Startup wizard" tool for every unit as you have access to all parameters of the slave units via the engineering interface of the fieldbus gateway. Doing so will adjust the inverter to the connected motor and, if required, the control loops will be adjusted to the load conditions of the application.

If available, you can load a matching parameter file to the inverter / servo inverter. It is important that the SBus address and in particular the SBus baud rate are not changed.

TIP

In particular with MOVIAXIS®, you have to check the communication settings of the IN-PDOs and OUT-PDOs. If the communication settings were changed by loading the parameter set, you can correct these settings by reloading the customized configuration or by executing the "autosetup" function again.

- MOVIAXIS®

  Process data communication is automatically configured in the axis module for each MOVIAXIS® slave unit if the gateway parameter "autosetup process data" for this unit is set to "ON". Only the process data objects required for communication between fieldbus gateway and MOVIAXIS® axis module are configured.

The unit-internal further processing of process data depends on the application and is not affected by the configuration by the fieldbus gateway.

After having configured the fieldbus gateway, you can set the parameters for the individual MOVIAXIS® axis modules. To do so, use the "PDO Editor" tool or "Parameter tree" to link the necessary IN and OUT PDOs used by the fieldbus gateway with the relevant control and status words.

- MOVIDRIVE® B and MOVITRAC® B

  Since the fieldbus gateway does not perform an automatic configuration for these inverters, you have to check the settings again as described in chapter "Configuring the fieldbus gateway and slave units".

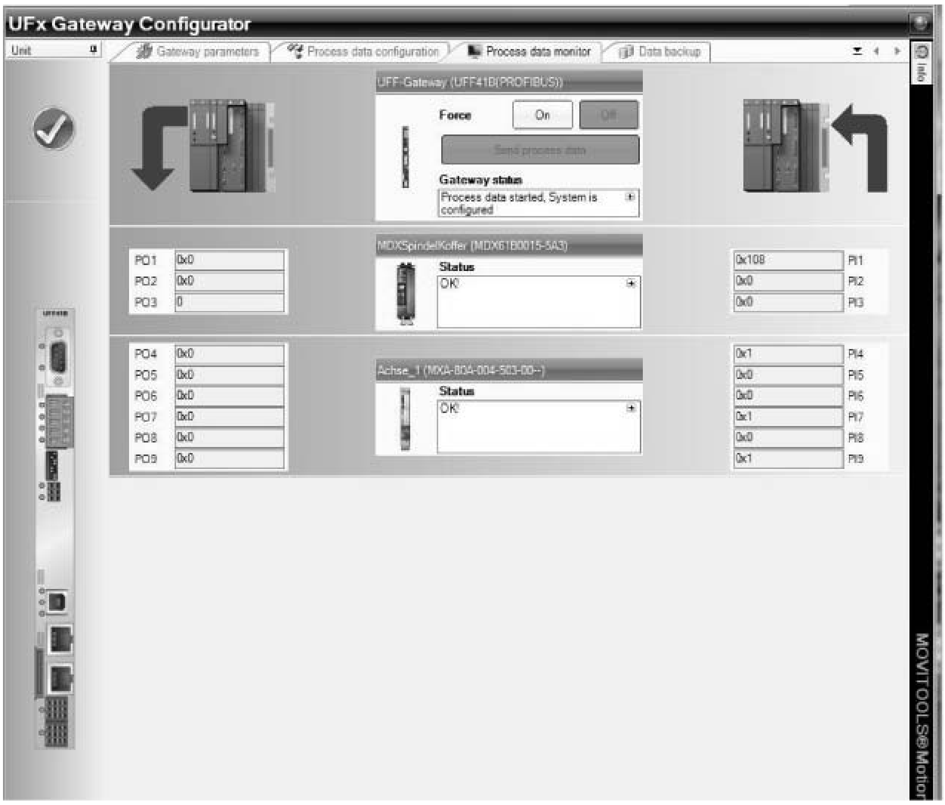
  Make sure that the following parameters are not changed when setting the inverter parameters to match your application:

  – P100 control signal source
  – P101 setpoint source
  – P880 / P890 SBus protocol
  – P881 / P891 SBus address
  – P884 / P894 SBus baud rate
  – P883 / P892 SBus timout interval
  – P889/899 parameter channel 2 = ON (only with MOVIDRIVE® B)

5.2.5 Monitoring and controlling process data

Process data diagnostics

In the UFx Gateway Configurator, open the "Process data monitor" tab (see following figure).


UFx Gateway Configurator
Unit
Gateway parameters
Process data configuration
Process data monitor
Data backup
Info
UFF-Gateway (UFF41B(PROFIBUS))
Force
On
MDXSpindelKoffer (MDX61B0015-5A3)
Achse_1 (MXA-80A-004-503-00--)
MOVITOOLS® Motion

Gateway status: Process data started, System is configured

MDXSpindelKoffer (MDX61B0015-5A3)

PO1 0x0
PO2 0x0
PO3 0

Status: OK!

0x108 PI1
0x0 PI2
0x0 PI3

Achse_1 (MXA-80A-004-503-00--)

PO4 0x0
PO5 0x0
PO6 0x0
PO7 0x0
PO8 0x0
PO9 0x0

Status: OK!

0x1 PI4
0x0 PI5
0x0 PI6
0x1 PI7
0x0 PI8
0x1 PI9

12084AEN

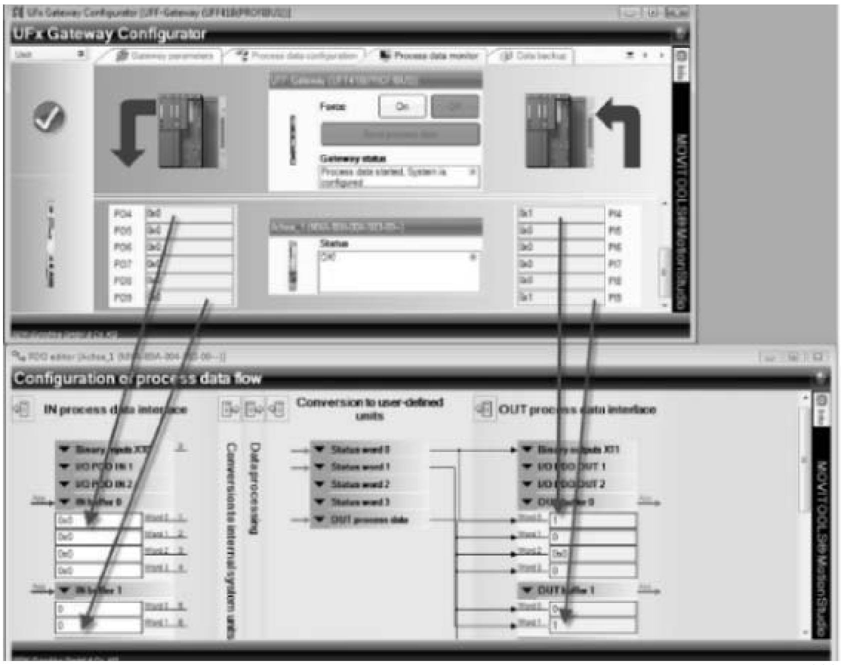
Check the data between fieldbus gateway and master controller. To apply different number formats to the individual numerical fields, make a right mouse click.

Checking process data in slave units

Do the following to check whether communication between fieldbus gateway and slave unit works properly:

- MOVIAXIS®

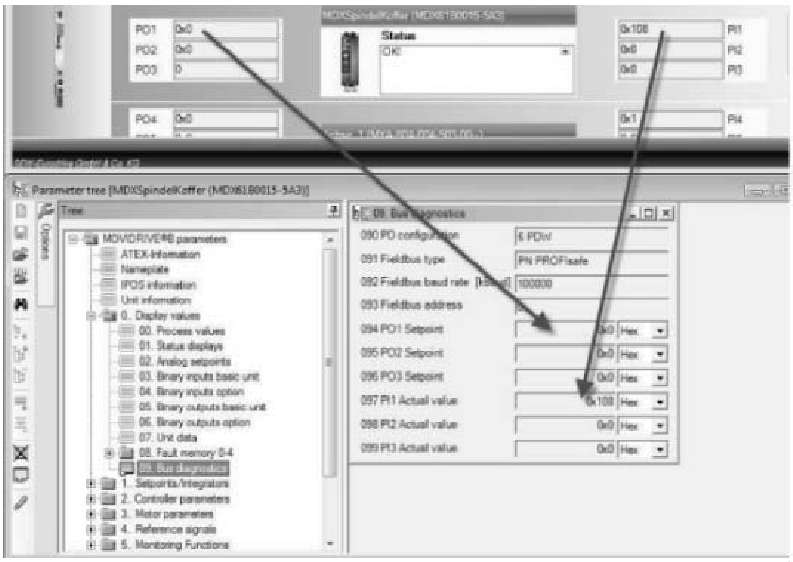
  You can use the PDO Editor to check process data. The input process data objects (IN-PDO) and output process data objects (OUT-PDO) are displayed (see following figure).



12085AEN

- MOVIDRIVE® B and MOVITRAC® B

  In MOVITOOLS® MotionStudio, you can check the process data using the "Parameter tree" tool in parameter group 09 "Bus diagnostics" (see following figure). The two tools "UFx Gateway Configurator" and "Parameter tree" can be open at the same time (see following figure).



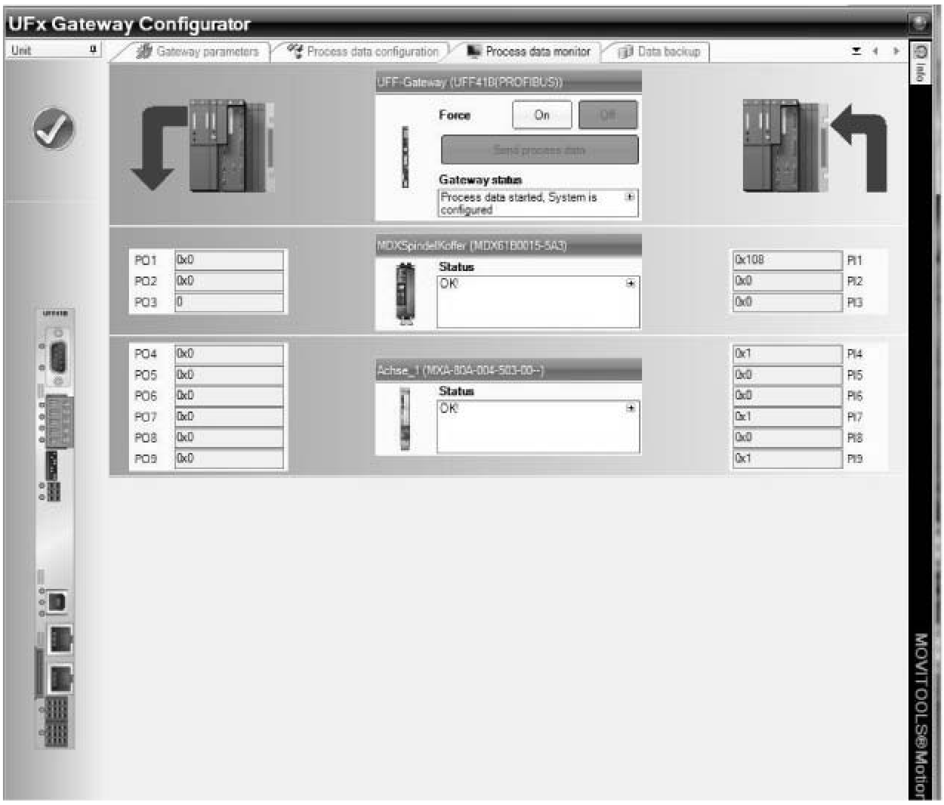
09. Bus diagnostics

090 PO configuration: 6 PDW
091 Fieldbus type: PN PROFIsafe
092 Fieldbus baud rate [kBaud]: 100000
093 Fieldbus address:
094 PO1 Setpoint: 0x0 Hex
095 PO2 Setpoint: 0x0 Hex
096 PO3 Setpoint: 0x0 Hex
097 PI1 Actual value: 0x108 Hex
098 PI2 Actual value: 0x0 Hex
099 PI3 Actual value: 0x0 Hex

12086AEN

Manual specification (forcing) of process output data

The process data monitor also lets you manually specify process output data without master controller (referred to as forcing).



UFF-Gateway (UFF41B(PROFIBUS))

Force: On / Off

Gateway status: Process data started, System is configured

MDXSpindelKoffer (MDX61B0015-5A3)

PO1	0x0	0x108	PI1
PO2	0x0	0x0	PI2
PO3	0	0x0	PI3

Status: OK!

Achse_1 (MXA-80A-004-503-00--)

PO4	0x0	0x1	PI4
PO5	0x0	0x0	PI5
PO6	0x0	0x0	PI6
PO7	0x0	0x1	PI7
PO8	0x0	0x0	PI8
PO9	0x0	0x1	PI9

Status: OK!

12084AEN

Activate force mode and enter the values in the now active fields. Clicking the "Send process data" button will send the entered values to the slave units via SBus instead of the valued received via fieldbus. Process input data cannot be specified manually.

5.2.6 Saving inverter data in the fieldbus gateway and using MOVITOOLS® MotionStudio

After having successfully configured the fieldbus gateway and after complete and verified parameterization of the slave units, the inverter parameters of the slave units can be saved on the SD card of the fieldbus gateway and on your PC using the project management of MOVITOOLS® MotionStudio.

Saving data on the SD memory card of the fieldbus gateway

To save the data of the slave units on the SD memory card of the fieldbus gateway, click on the [Data backup] tab in the UFx Gateway Configurator and click the [Data backup] button. For this purpose, all drives must be at standstill and process data communication must be stopped.

Clicking the [Data backup] button of the displayed slave units will copy the parameter set of this unit to the SD card of the fieldbus gateway.

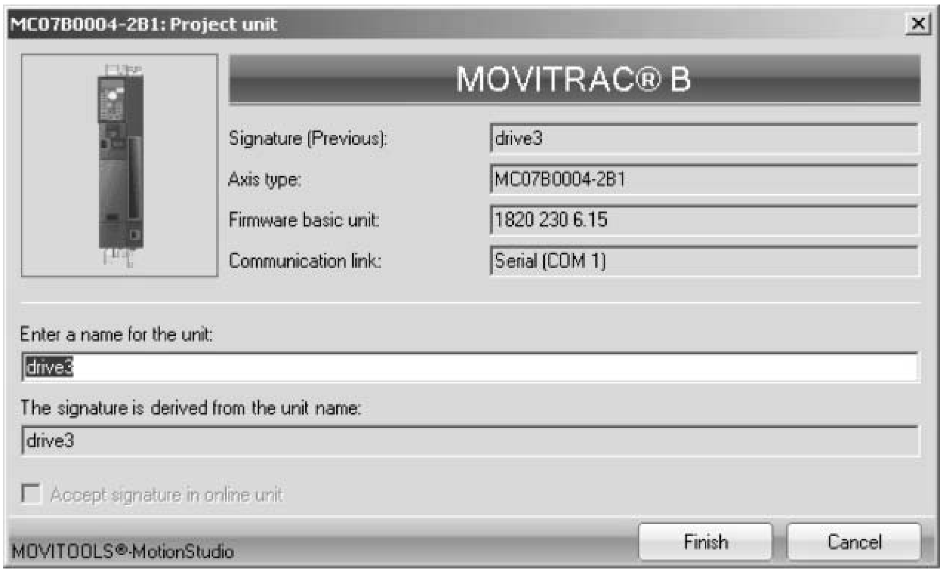
Setting the "Automatic update" function to "OFF" disables the restore function for this unit after unit replacement (see also chapter 5.1.5).

Saving data using the project management in MOVITOOLS® MotionStudio

Proceed as follows to configure existing units in the network:

1. Switch to the network view with the "Network view" tab.
2. Perform a unit scan.

   This will display all units that are physically connected and accessible online.
3. Select the unit you want to configure.
4. Drag the scanned unit from the network view into project view (drag and drop) or select the [Project unit] command from the context menu.



MC07B0004-2B1: Project unit

MOVITRAC® B

Signature (Previous): drive3

Axis type: MC07B0004-2B1

Firmware basic unit: 1820 230 6.15

Communication link: Serial (COM 1)

Enter a name for the unit: drive3

The signature is derived from the unit name: drive3

□ Accept signature in online unit

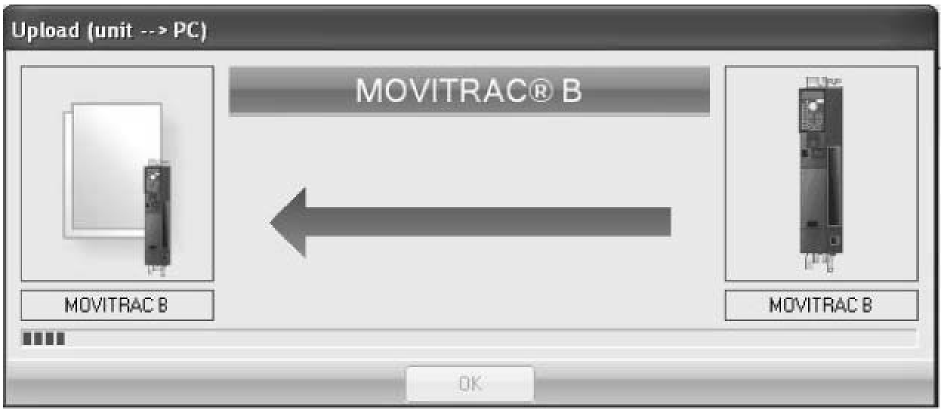
12116AEN

   This opens the "Project unit" window.
5. Use the name (signature) of the unit that is accessible online.

	TIP
ℹ	Proceed as follows if you do **NOT** want to transfer the name (signature) from the unit that is available online: • Enter a new signature. • Activate the "Accept signature in online unit" control field. Doing so ensures that the unit can be clearly identified in the future.

6. Click [Finish].

   The parameters are then transferred from the unit, which can be accessed online, to the working memory.



12117AEN

7. Confirm with [OK].

   The mini symbol on the unit node will then disappear in the network view.

8. Save your project.

   The parameter is then transferred from the working memory to the parameter file where it is permanently saved.

5.2.7 Error processing and status messages

The fieldbus gateway distinguishes between status and error messages of the fieldbus gateway and individual slave units. For every slave, a status word is stored in an individual parameter. The following table gives an overview of the assignment of individual bits of the slave status word.

Status word slave	
Bit	**Assignment**
2	Slave timeout
3	Configuration error in project planning
4	Configuration error in process data
5	Update error
9	Data backup
10	Update in progress
11	Replaced axis detected
15	Error while saving data
17	Error while reading UUID during data backup
30	Unit update after timeout

The status of the fieldbus gateway is stored in a parameter in bit code. The following table gives an overview of the assignment of individual bits of the fieldbus gateway status word. The fieldbus gateway status results from ORing the bits in the individual slave states if the bit assignment in the slave and fieldbus gateway states corresponds.

Fieldbus gateway status word	
Bit	**Assignment**
0	Malfunction
1	Fieldbus timeout
2	Slave timeout
3	Configuration error in project planning
4	Configuration error in process data
5	Update error
6	Process data started
7	Process data stopped
8	Configured
9	Data backup
10	Update in progress
11	Replaced axis detected
12	Bus scan
13	Autosetup slaves
14	SBus initialization
15	Error during data backup
30	Unit update after timeout

This allows for detailed error diagnostics. For example, if the fieldbus gateway indicates a configuration error during configuration (bit 3), the slave where this error has occurred can be determined from the status of the slaves. Bits, which indiate an error, are reset with an error reset (bits 0 - 5, bit 11, bit 15, bit 30).

Communication error between fieldbus gateway and slave unit

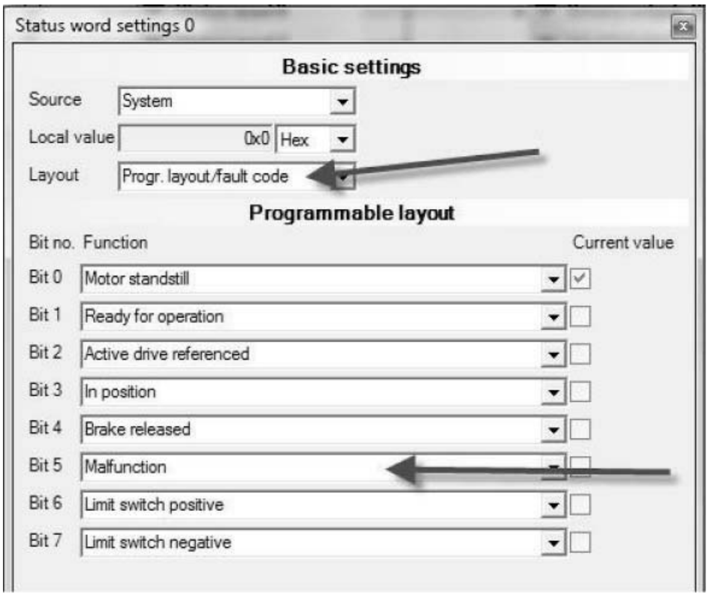
If the fieldbus gateway detects a timeout during communcation with a slave unit, then the fieldbus gateway automatically shows fault number F111 in the first word of the process image of the relevant slave unit.

A timeout is detected by monitoring the process data communication between fieldbus gateway and slave. A communication error is automatically reset as soon as the malfunction is eliminated.

The following parameters must be set in these units to enable the fieldbus gateway to signal error states of connected units to the master controller:

- MOVIDRIVE® B, MOVITRAC® B

  P873 = Status word 1 or status word 3
- MOVIAXIS® (see following figure)



Status word settings 0

Basic settings

Source: System

Local value: 0x0 Hex

Layout: Progr. layout/fault code

Programmable layout

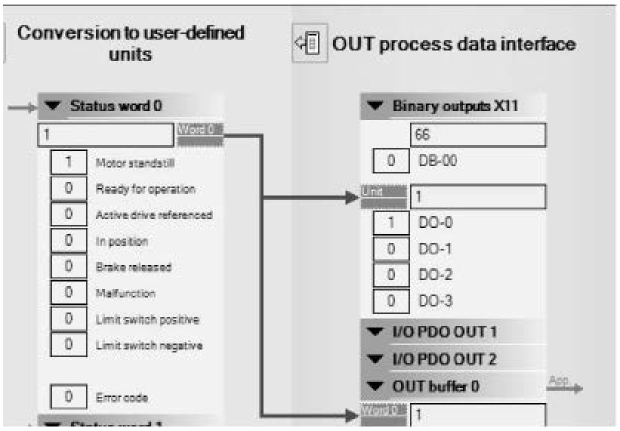
Bit no.	Function	Current value
Bit 0	Motor standstill	☑
Bit 1	Ready for operation	□
Bit 2	Active drive referenced	□
Bit 3	In position	□
Bit 4	Brake released	□
Bit 5	Malfunction	□
Bit 6	Limit switch positive	□
Bit 7	Limit switch negative	□

12108AEN

Status word settings:

– Selection field "Layout": Progr. layout/fault code
– Selection field "Bit 5: Malfunction"

This status word is linked with the corresponding output process data object (see following figure).



Conversion to user-defined units

Status word 0: 1 (Word 0)

Value	Function
1	Motor standstill
0	Ready for operation
0	Active drive referenced
0	In position
0	Brake released
0	Malfunction
0	Limit switch positive
0	Limit switch negative
0	Error code

Status word 1

OUT process data interface

Binary outputs X11: 66

Value	Output
0	DB-00

Unit: 1

Value	Output
1	DO-0
0	DO-1
0	DO-2
0	DO-3

I/O PDO OUT 1

I/O PDO OUT 2

OUT buffer 0 (App)

Word 0: 1

12109AEN

TIP

The fieldbus gateway does not verify correct parameter setting of the status word. A deviating parameter setting will cause the controller to not correctly detect either communication timeouts with the slave units or other errors.

Fieldbus timeout

The fieldbus gateway detects a failed communication with the master controller (fieldbus master). In this case, the fieldbus gateway sends "0" signals to all slave units in their process image and in this way stops all drives using the set rapid stop ramp. Fieldbus communication will automatically be resumed after a fieldbus timeout.

Used CAN IDs

The following CAN IDs are used for communication between fieldbus gateway and slave units.

Number of		Calculation of CAN IDs of	
process data per drive	**CAN telegrams**	**process inputs PI**	**process outputs PO**
3 process data for MOVIDRIVE® B and MOVITRAC® B	1 to 3 CAN telegrams	8 x SBus address + 3	8 x SBus address + 4
1 to 4 process data for MOVIAXIS®	1 CAN telegram	8 x SBus address + 3	8 x SBus address + 0
5 to 8 process data for MOVIAXIS®	2 CAN telegrams	1. CAN telegram: 8 x SBus address + 3 2. CAN telegram: 8 x SBus address + 4	1. CAN telegram: 8 x SBus address + 0 2. CAN telegram: 8 x SBus address + 1
9 to 12 process data for MOVIAXIS®	3 CAN telegrams	1. CAN telegram: 8 x SBus address + 3 2. CAN telegram: 8 x SBus address + 4 3. CAN telegram: 8 x SBus address + 5	1. CAN telegram: 8 x SBus address + 0 2. CAN telegram: 8 x SBus address + 1 3. CAN telegram: 8 x SBus address + 2
13 to 16 process data for MOVIAXIS®	4 CAN telegrams	1. CAN telegram: 8 x SBus address + 3 2. CAN telegram: 8 x SBus address + 4 3. CAN telegram: 8 x SBus address + 5 4. CAN telegram: 8 x SBus address + 7	1. CAN telegram: 8 x SBus address + 0 2. CAN telegram: 8 x SBus address + 1 3. CAN telegram: 8 x SBus address + 2 4. CAN telegram: 8 x SBus address + 6

TIPS

A sychronization telegram is also transmitted to ensure data consistency:

SyncID for CAN 1 and CAN 2 = 1

This calculation directive ensures the consistency of IDs calculated for MOVIAXIS® using the "Single-axis positioning" technology editor.

6 Configuration and Startup (EtherNet/IP)

This chapter provides you with information on the configuration of the EtherNet/IP master and startup of the fieldbus gateway for fieldbus operation. Prerequisite is the correct connection and setting of the IP address parameters of UFR41B as described in the "Assembly and Installation Instructions" chapter.

6.1 Validity of the EDS file for UFR41B

ℹ	**TIP**
	Do not edit or amend the entries in the EDS file. SEW assumes no liability for inverter malfunctions caused by a modified EDS file!

SEW-EURODRIVE provides the following EDS file for projecting the scanner (EtherNet/IP master):

- SEW_GATEWAY_UFR41B.eds

ℹ	**TIP**
	Current versions of the EDS files for the UFR41B option are available on the SEW homepage under the heading "Software".

6.2 Configuring the master (EtherNet/IP scanner)

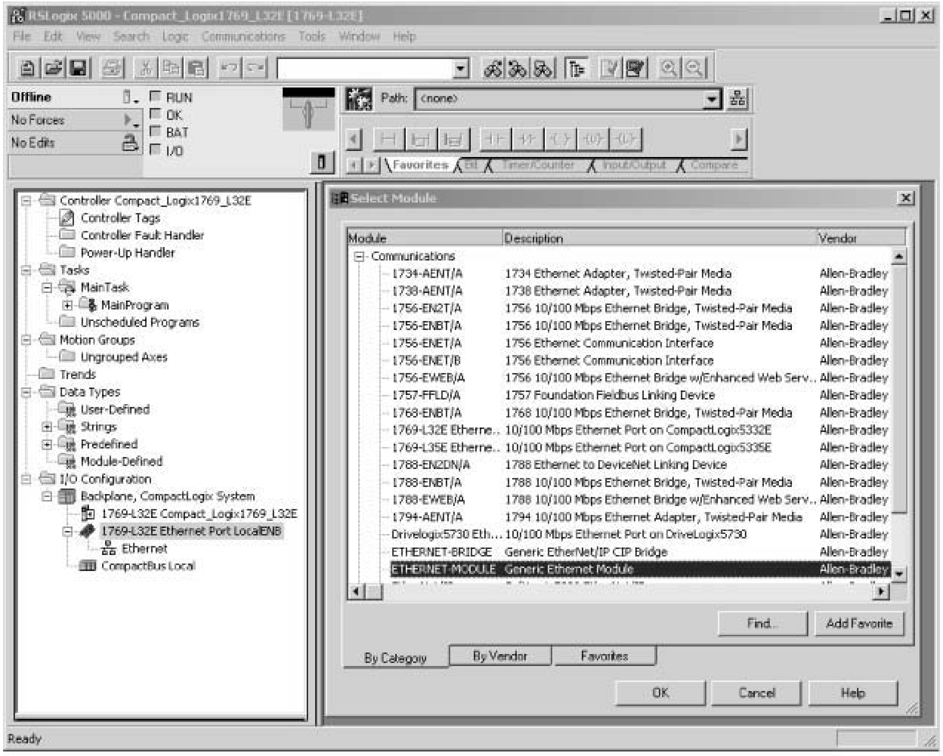
The following example refers to the configuration of the AllenBradley CompactLogix 1769-L32E controller with RSLogix 5000 programming software. The EtherNet/IP interface is already integrated in the CPU component.

TIP

If a CPU without an EtherNet/IP interface is used, an Ethernet communication interface must first be added to the IO configuration.

Process data exchange

In the following project planning example, the UFR41B option is added to a project. To do so, go to the view "Controller Organizer" in the RSLogix 5000 program as shown in the screenshot below (use the tree structure on the left side of the screen).

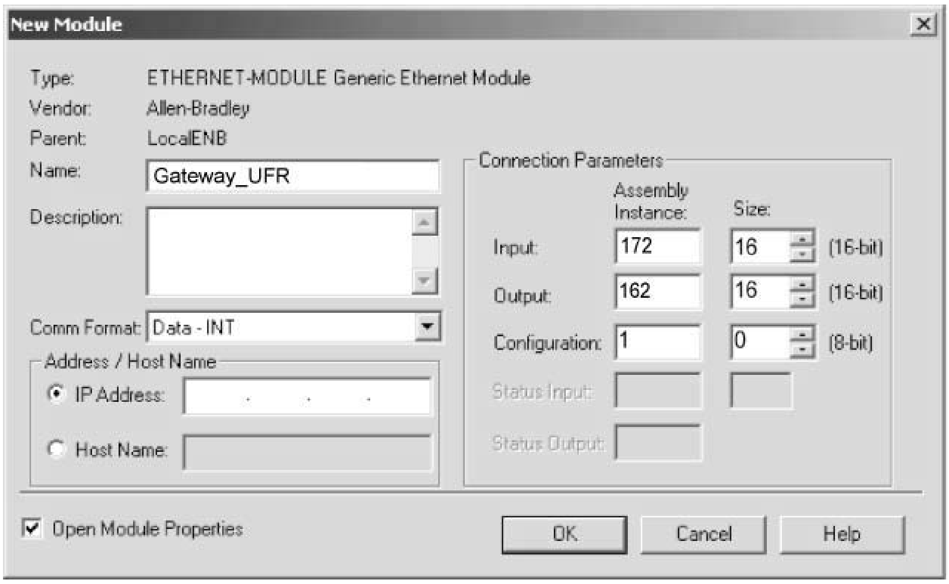


Module	Description	Vendor
Communications		
1734-AENT/A	1734 Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
1738-AENT/A	1738 Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
1756-EN2T/A	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-ENBT/A	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-ENET/A	1756 Ethernet Communication Interface	Allen-Bradley
1756-ENET/B	1756 Ethernet Communication Interface	Allen-Bradley
1756-EWEB/A	1756 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv...	Allen-Bradley
1757-FFLD/A	1757 Foundation Fieldbus Linking Device	Allen-Bradley
1768-ENBT/A	1768 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1769-L32E Etherne...	10/100 Mbps Ethernet Port on CompactLogix5332E	Allen-Bradley
1769-L35E Etherne...	10/100 Mbps Ethernet Port on CompactLogix5335E	Allen-Bradley
1788-EN2DN/A	1788 Ethernet to DeviceNet Linking Device	Allen-Bradley
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv...	Allen-Bradley
1794-AENT/A	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
Drivelogix5730 Eth...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley

11709AXX

- In the "I/O Configuration" folder, select the entry "1769-L32E Ethernet Port LocalENB" as the Ethernet communication interface. Make a right mouse click to open the context menu and choose "New Module". The selection window "Select Module Type" appears.
- To add the UFR41B option to the project, select the entry "ETHERNET MODULE" from the category "Communications". Confirm your selection by clicking [OK].
- The "New Module" window opens.

First enter the name under which the data is stored in the controller tags for the newly created module, and then enter the IP address.



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: Gateway_UFR
Description:
Comm Format: Data - INT

Address / Host Name
● IP Address: . . .
○ Host Name:

Connection Parameters

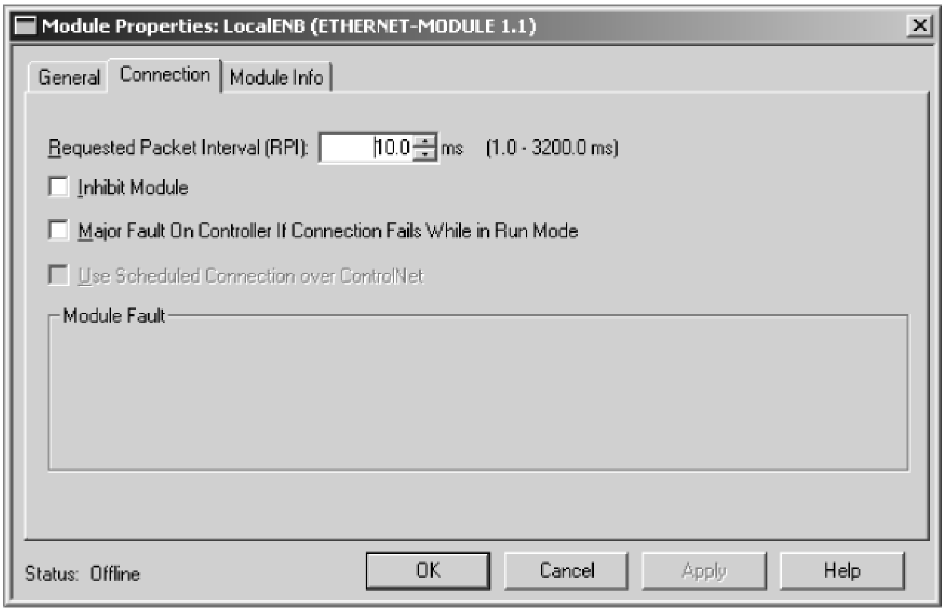
	Assembly Instance:	Size:	
Input:	172	16	(16-bit)
Output:	162	16	(16-bit)
Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

12127AXX

- **For the data format, open the dropdown menu "Comm-Format"and choose the entry "Data - INT". Process data for UFR41B always contains 16 bits (INT).**
- In the "Connection Parameters" group box, enter the value "172" in the "Input Assembly Instance" input field. The input data of the PLC must be linked to the output instance of UFR41B .
- To establish a controlling connection, in the "Connection Parameters" group box, enter the value "162" in the "Output Assembly Instance" input field. The input data of the PLC must be linked to the output instance of UFR41B .
- In the selection fields "Input Size" and "Output Size," set a maximum value of "64" (16 bit) as the data length.
- In the "Configuration Size" selection field, enter the value "0." The "Configuration Assembly Instance" is not used in this case.
- Click [OK] to complete the process.
- To ensure compatibility with existing DeviceNet configurations, you can also choose the data type "SINT" in the "Comm Format" selection field. In this case, you must ensure that an even number of bytes (2 - 128) is configured and that data consistency is maintained during operation when the IO data is accessed.

Other settings The "Connection" tab page is used to set the data rate and, if required, the error response of the controller.



Module Properties: LocalENB (ETHERNET-MODULE 1.1)

Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

□ Use Scheduled Connection over ControlNet

Status: Offline

11712AXX

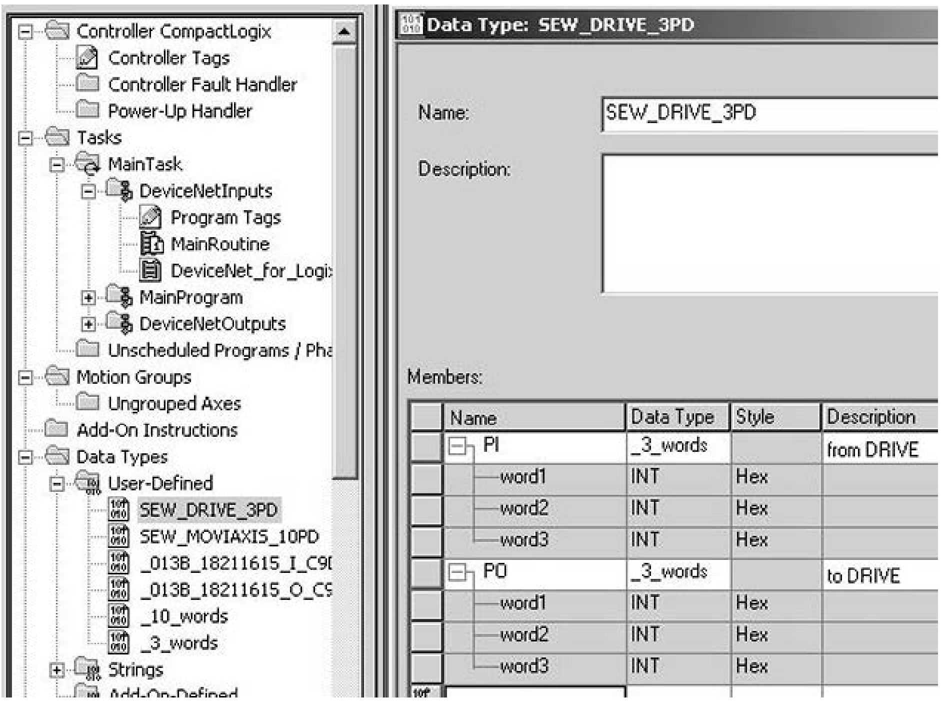
- The UFR41B option supports a minimum data rate (input field "Requested Packet Interval (RPI)") of 4 ms. Longer cycle times can be implemented without any problems.
- Click [OK]. You have now configured process data exchange with a UFR41B.

6.3 *Project planning examples in RSLogix 5000*

6.3.1 UFR41B fieldbus gateway with 16 PD data exchange

1. Set the IP address of the DFE33B option see chapter "Setting the IP address parameters").
2. Add the UFR41B fieldbus gateway to the EtherNet/IP configuration as described in chapter 5.2.
3. You can now start integration into the RSLogix project.

   To do so, create a controller tag with a user-defined data type to create a simple, data consistent interface to the process data of UFR41B (see following figure).



- Controller CompactLogix
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - DeviceNetInputs
      - Program Tags
      - MainRoutine
      - DeviceNet_for_Logi:
    - MainProgram
    - DeviceNetOutputs
  - Unscheduled Programs / Pha
- Motion Groups
  - Ungrouped Axes
- Add-On Instructions
- Data Types
  - User-Defined
    - SEW_DRIVE_3PD
    - SEW_MOVIAXIS_10PD
    - _013B_18211615_I_C9(
    - _013B_18211615_O_C9
    - _10_words
    - _3_words
  - Strings
  - Add-On-Defined

Data Type: SEW_DRIVE_3PD

Name: SEW_DRIVE_3PD

Description:

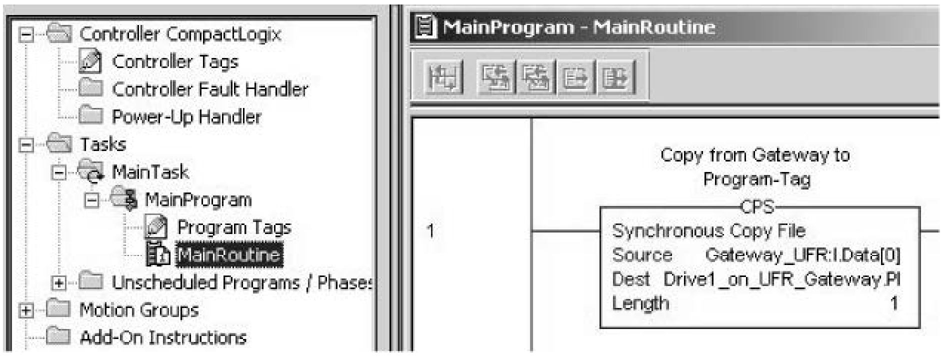
Members:

Name	Data Type	Style	Description
PI	_3_words		from DRIVE
word1	INT	Hex	
word2	INT	Hex	
word3	INT	Hex	
PO	_3_words		to DRIVE
word1	INT	Hex	
word2	INT	Hex	
word3	INT	Hex	

11962AXX

The description for the process input and output data of the controller tag can match the definition of the process data (PD) in the inverters.

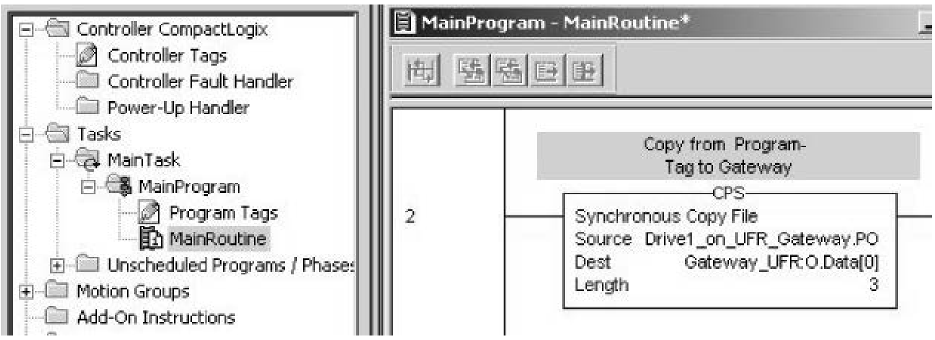
4. To copy the data of the UFR41B fieldbus gateway into the new data structure, a CPS command is inserted at the beginning of the "MainRoutine". This command reads the data from the controller tag beginning from the offset (see figure below).



Copy from Gateway to Program-Tag
CPS
Synchronous Copy File
Source Gateway_UFR:I.Data[0]
Dest Drive1_on_UFR_Gateway.PI
Length 1

12129AXX

To copy the data from the new data structure to the UFR41B fieldbus gateway, insert a CPS command at the end of the "MainRoutine" (see following figure). The data of the individual drives are written to the fieldbus gateway in the data structure and with the corresponding offset.



Copy from Program-Tag to Gateway
CPS
Synchronous Copy File
Source Drive1_on_UFR_Gateway.PO
Dest Gateway_UFR:O.Data[0]
Length 3

12130AXX

5. Now save the project and transfer it to the PLC. The PLC is set to RUN mode.

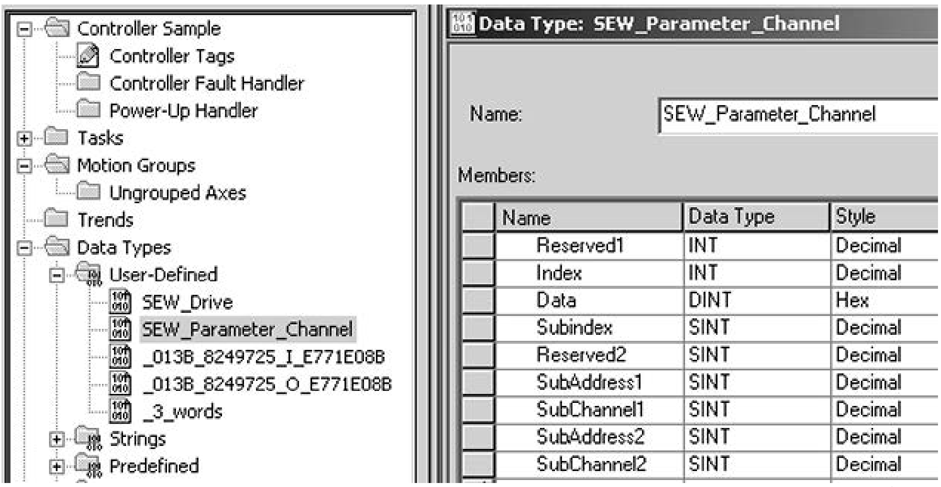
   You can now read the actual values from the gateway configurator and write setpoint values.

   The process data should correspond with the values displayed in the Gateway Configurator of MOVITOOLS® MotionStudio (see following figure).

6.3.2 Access to UFR41B fieldbus gateway parameters

In order to get an easy-to-use read access to parameters of the UFR41B fieldbus gateway via *explicit messages* and the *register object*, follow these steps:

1. Create a user-defined data structure "SEW_Parameter_Channel" (see following figure)



- Controller Sample
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
- Motion Groups
  - Ungrouped Axes
- Trends
- Data Types
  - User-Defined
    - SEW_Drive
    - SEW_Parameter_Channel
    - _013B_8249725_I_E771E08B
    - _013B_8249725_O_E771E08B
    - _3_words
  - Strings
  - Predefined

Data Type: SEW_Parameter_Channel

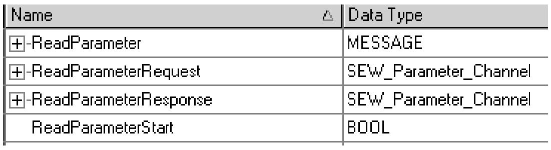
Name: SEW_Parameter_Channel

Members:

Name	Data Type	Style
Reserved1	INT	Decimal
Index	INT	Decimal
Data	DINT	Hex
Subindex	SINT	Decimal
Reserved2	SINT	Decimal
SubAddress1	SINT	Decimal
SubChannel1	SINT	Decimal
SubAddress2	SINT	Decimal
SubChannel2	SINT	Decimal

11764AXX

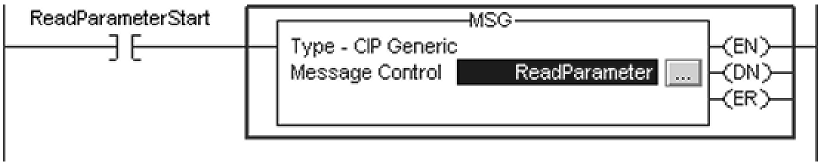
2. Define the following controller tags (see figure below).



Name	Data Type
+-ReadParameter	MESSAGE
+-ReadParameterRequest	SEW_Parameter_Channel
+-ReadParameterResponse	SEW_Parameter_Channel
ReadParameterStart	BOOL

11765AXX

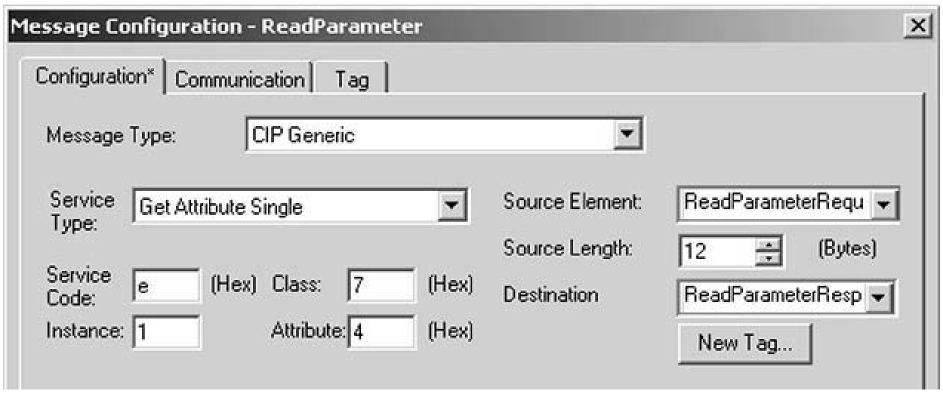
3. Create a rung for the "ReadParameter" execution (following figure).



11766AXX

- For contact, select the tag "ReadParameterStart"
- For the Message Control, select the tag "ReadParameter"

4. Click on [...] in the MSG instruction to open the "Message Configuration" window (see following figure).



Message Configuration - ReadParameter

Configuration* | Communication | Tag

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 7 (Hex)

Instance: 1 Attribute: 4 (Hex)

Source Element: ReadParameterRequ

Source Length: 12 (Bytes)

Destination: ReadParameterResp

New Tag...

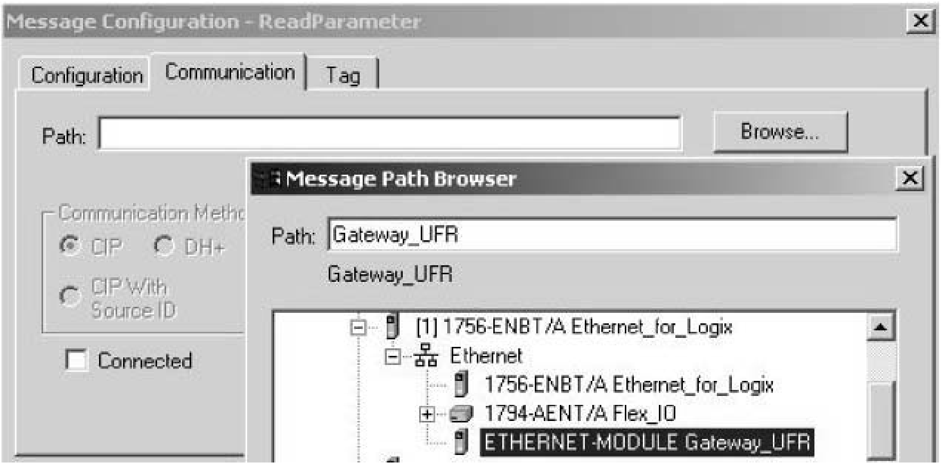
11767AXX

Select "CIP Generic" as "message type". Fill the other fields in the following order:

A Source Element = ReadParameterRequest.Index
B Source Length = 12
C Destination = ReadParameterResponse.Index
D Class = 7$_{hex}$
E Instance = 1
F Attribute = 4$_{hex}$
G Service Code = e$_{hex}$

The service type is set automatically.

5. Specify the target device on the "Communication" tab. Click the [Browse] button and select the required unit from the IO configuration (under Ethernet) in the Message Path Browser (see following figure).



Message Configuration - ReadParameter

Configuration | Communication | Tag

Path:

Browse...

Communication Metho
● CIP ○ DH+
○ CIP With Source ID

□ Connected

Message Path Browser

Path: Gateway_UFR

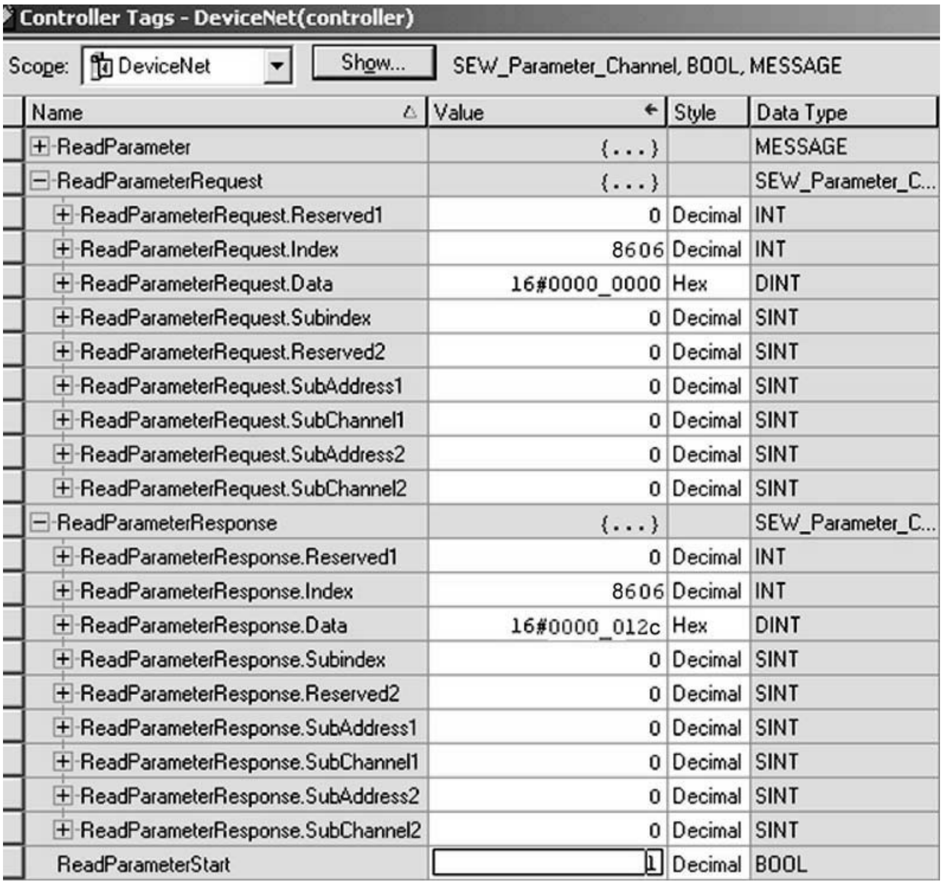
Gateway_UFR

[1] 1756-ENBT/A Ethernet_for_Logix
Ethernet
1756-ENBT/A Ethernet_for_Logix
1794-AENT/A Flex_IO
ETHERNET-MODULE Gateway_UFR

12131AXX

Do not select the "Connected" checkbox because both the controller and the UFR41B option permit only a limit number of connections.

6. After downloading the changes to the PLC, the index of the parameter to be read can be entered at *ReadParameterRequest.Index*. By altering *ReadParameterStart* to "1" the read request is executed once (see following figure).



Controller Tags - DeviceNet(controller)

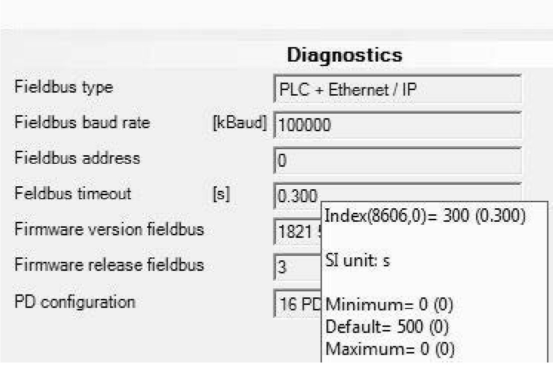
Scope: DeviceNet Show... SEW_Parameter_Channel, BOOL, MESSAGE

Name	Value	Style	Data Type
⊞ ReadParameter	{...}		MESSAGE
⊟ ReadParameterRequest	{...}		SEW_Parameter_C...
⊞ ReadParameterRequest.Reserved1	0	Decimal	INT
⊞ ReadParameterRequest.Index	8606	Decimal	INT
⊞ ReadParameterRequest.Data	16#0000_0000	Hex	DINT
⊞ ReadParameterRequest.Subindex	0	Decimal	SINT
⊞ ReadParameterRequest.Reserved2	0	Decimal	SINT
⊞ ReadParameterRequest.SubAddress1	0	Decimal	SINT
⊞ ReadParameterRequest.SubChannel1	0	Decimal	SINT
⊞ ReadParameterRequest.SubAddress2	0	Decimal	SINT
⊞ ReadParameterRequest.SubChannel2	0	Decimal	SINT
⊟ ReadParameterResponse	{...}		SEW_Parameter_C...
⊞ ReadParameterResponse.Reserved1	0	Decimal	INT
⊞ ReadParameterResponse.Index	8606	Decimal	INT
⊞ ReadParameterResponse.Data	16#0000_012c	Hex	DINT
⊞ ReadParameterResponse.Subindex	0	Decimal	SINT
⊞ ReadParameterResponse.Reserved2	0	Decimal	SINT
⊞ ReadParameterResponse.SubAddress1	0	Decimal	SINT
⊞ ReadParameterResponse.SubChannel1	0	Decimal	SINT
⊞ ReadParameterResponse.SubAddress2	0	Decimal	SINT
⊞ ReadParameterResponse.SubChannel2	0	Decimal	SINT
ReadParameterStart	1	Decimal	BOOL

11966BXX

On response to the read request, *ReadParameterResponse.Index* should indicate the read index and *ReadParameterResponse.Data* should contain the read data. In this example, the timeout interval of the UFR41B fieldbus gateway (index 8606) set by the scanner has been read (012Chex ≙ 0.3 s).

You can check the value in the MOVITOOLS® MotionStudio parameter tree (see figure below). The tooltip of a parameter displays for example index, subindex, factor, etc. of the parameter.



Diagnostics

Fieldbus type		PLC + Ethernet / IP
Fieldbus baud rate	[kBaud]	100000
Fieldbus address		0
Feldbus timeout	[s]	0.300
Firmware version fieldbus		1821
Firmware release fieldbus		3
PD configuration		16 PD

Index(8606,0)= 300 (0.300)

SI unit: s

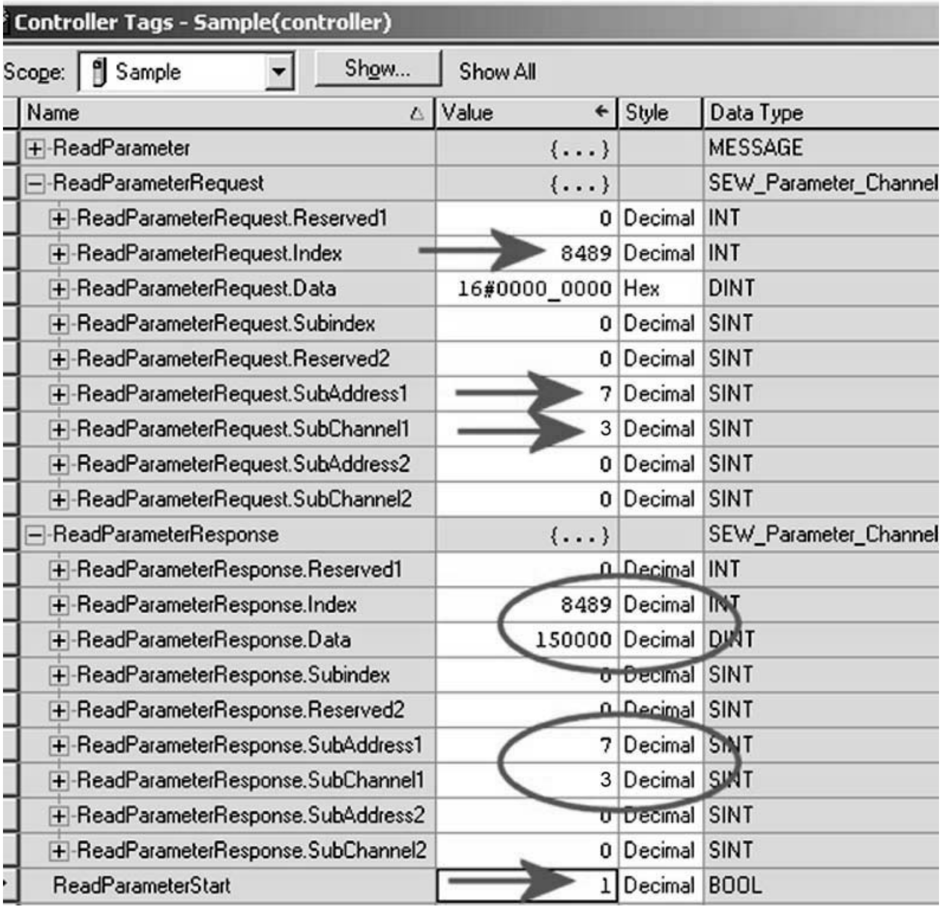
Minimum= 0 (0)
Default= 500 (0)
Maximum= 0 (0)

12061AXX

6.3.3 Access to unit parameters of lower-level units

Access to the device parameters of, for example, a MOVITRAC® B connected to the CAN 1 system bus of the UFR41B fieldbus gateway is the same as access to the device parameters of the UFR41B fieldbus gateway itself (see chapter 5.4.2)

The only difference is that **Read/WriteParameterRequest.SubChannel1** must be set to **3** and **Read/WriteParameterRequest.SubAddress1** must be set to the SBus address of MOVITRAC® B connected to UFR41B (see figure below).



Controller Tags - Sample(controller)

Scope: Sample Show... Show All

Name	Value	Style	Data Type
⊞ ReadParameter	{...}		MESSAGE
⊟ ReadParameterRequest	{...}		SEW_Parameter_Channel
⊞ ReadParameterRequest.Reserved1	0	Decimal	INT
⊞ ReadParameterRequest.Index	8489	Decimal	INT
⊞ ReadParameterRequest.Data	16#0000_0000	Hex	DINT
⊞ ReadParameterRequest.Subindex	0	Decimal	SINT
⊞ ReadParameterRequest.Reserved2	0	Decimal	SINT
⊞ ReadParameterRequest.SubAddress1	7	Decimal	SINT
⊞ ReadParameterRequest.SubChannel1	3	Decimal	SINT
⊞ ReadParameterRequest.SubAddress2	0	Decimal	SINT
⊞ ReadParameterRequest.SubChannel2	0	Decimal	SINT
⊟ ReadParameterResponse	{...}		SEW_Parameter_Channel
⊞ ReadParameterResponse.Reserved1	0	Decimal	INT
⊞ ReadParameterResponse.Index	8489	Decimal	INT
⊞ ReadParameterResponse.Data	150000	Decimal	DINT
⊞ ReadParameterResponse.Subindex	0	Decimal	SINT
⊞ ReadParameterResponse.Reserved2	0	Decimal	SINT
⊞ ReadParameterResponse.SubAddress1	7	Decimal	SINT
⊞ ReadParameterResponse.SubChannel1	3	Decimal	SINT
⊞ ReadParameterResponse.SubAddress2	0	Decimal	SINT
⊞ ReadParameterResponse.SubChannel2	0	Decimal	SINT
ReadParameterStart	1	Decimal	BOOL

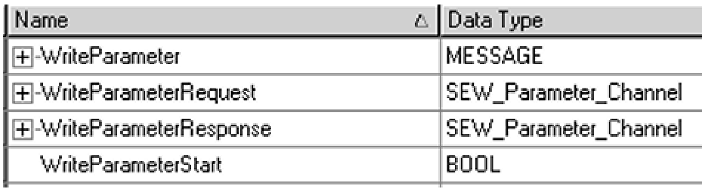
11775BXX

In this example, MOVITRAC® B connected to a CAN 1 system bus of the UFR41B option with SBus address 7 read the value 150 rpm from *P160 Fixed setpoint n11* (index 8489).

For a schematic representation of the parameter access to lower-level units, refer to the chapter "Appendix".

Only a few additions are necessary for activating write access to a parameter.

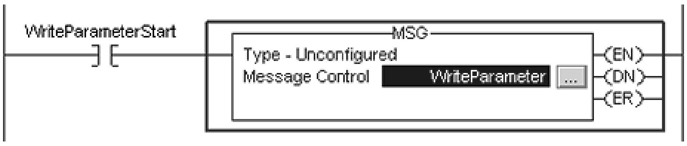
- Create the controller tags (see following figure).



Name	Data Type
⊞-WriteParameter	MESSAGE
⊞-WriteParameterRequest	SEW_Parameter_Channel
⊞-WriteParameterResponse	SEW_Parameter_Channel
WriteParameterStart	BOOL

11771AXX

- Create a rung for executing the "WriteParameter" command (see following figure).

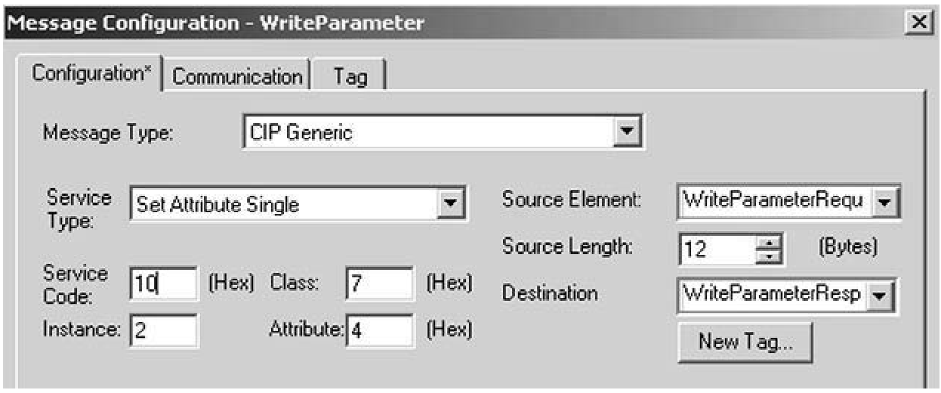


11772AXX

For contact, select the tag "WriteParameterStart"

For message control, select the tag "WriteParameter"

- Click on ... in the MSG instruction to open the "Message Configuration" window (see following figure).



Message Configuration - WriteParameter

Configuration* | Communication | Tag

Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex) Class: 7 (Hex)

Instance: 2 Attribute: 4 (Hex)

Source Element: WriteParameterRequ

Source Length: 12 (Bytes)

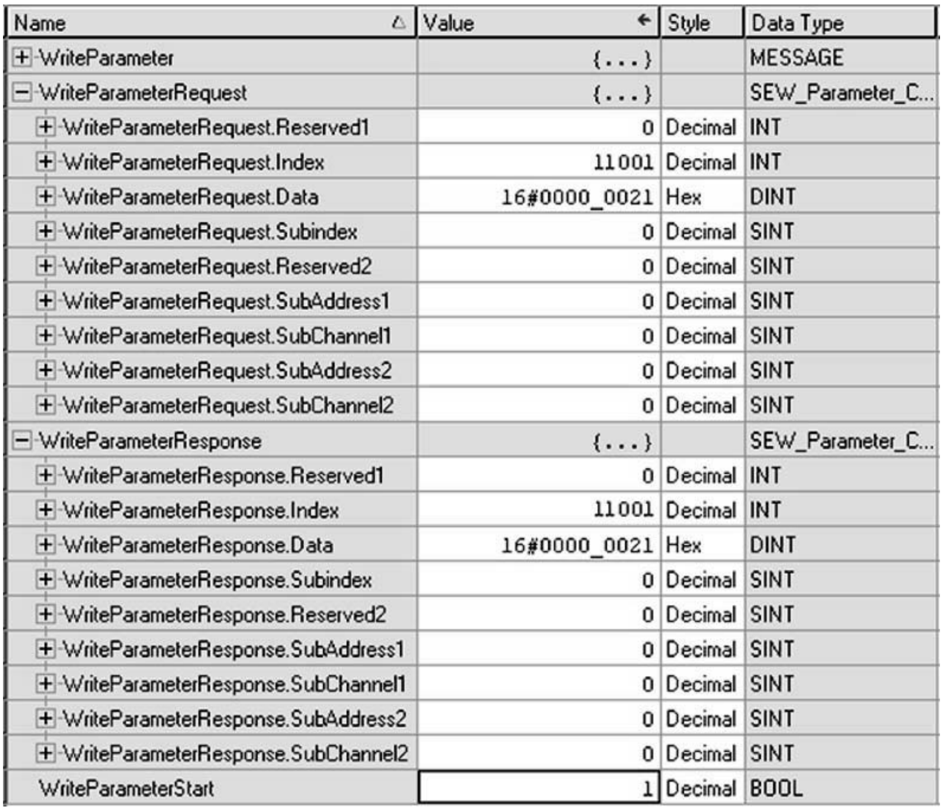
Destination: WriteParameterResp

11773AXX

Fill the other fields in the following sequence:

- Source Element = WriteParameterRequest.Index
- Source Length = 12
- Destination = WriteParameterResponse.Index
- Class = 7_{hex}
- Instance = 2
- Attribute = 4_{hex}
- Service Code = 10_{hex}

- After downloading the changes to the PLC, index and value to be written into the parameter can be entered at *WriteParameterRequest.Index* and *WriteParameterRequest.Data.* By altering *WriteParameterStart* to "1", the write request is executed once (see following figure).



Name	Value	Style	Data Type
⊞ WriteParameter	{...}		MESSAGE
⊟ WriteParameterRequest	{...}		SEW_Parameter_C...
⊞ WriteParameterRequest.Reserved1	0	Decimal	INT
⊞ WriteParameterRequest.Index	11001	Decimal	INT
⊞ WriteParameterRequest.Data	16#0000_0021	Hex	DINT
⊞ WriteParameterRequest.Subindex	0	Decimal	SINT
⊞ WriteParameterRequest.Reserved2	0	Decimal	SINT
⊞ WriteParameterRequest.SubAddress1	0	Decimal	SINT
⊞ WriteParameterRequest.SubChannel1	0	Decimal	SINT
⊞ WriteParameterRequest.SubAddress2	0	Decimal	SINT
⊞ WriteParameterRequest.SubChannel2	0	Decimal	SINT
⊟ WriteParameterResponse	{...}		SEW_Parameter_C...
⊞ WriteParameterResponse.Reserved1	0	Decimal	INT
⊞ WriteParameterResponse.Index	11001	Decimal	INT
⊞ WriteParameterResponse.Data	16#0000_0021	Hex	DINT
⊞ WriteParameterResponse.Subindex	0	Decimal	SINT
⊞ WriteParameterResponse.Reserved2	0	Decimal	SINT
⊞ WriteParameterResponse.SubAddress1	0	Decimal	SINT
⊞ WriteParameterResponse.SubChannel1	0	Decimal	SINT
⊞ WriteParameterResponse.SubAddress2	0	Decimal	SINT
⊞ WriteParameterResponse.SubChannel2	0	Decimal	SINT
WriteParameterStart	1	Decimal	BOOL

11967BXX

On response to the write request, *WriteParameterResponse.Index* should give the written index and *WriteParameterResponse.Data* should contain the written data. In this example, 21hex (33 dec) was written to index 11001 (H1).

You can check the value in MOVITOOLS® MotionStudio.

7 Ethernet Industrial Protocol (EtherNet/IP)

7.1 Introduction

The EtherNet Industrial Protocol (EtherNet/IP) is an open communication standard based on the classic EtherNet protocols TCP/IP and UDP/IP.

EtherNet/IP has been defined by the **O**pen **D**eviceNet **V**endor **A**ssociation (ODVA) and **C**ontrolNet **I**nternational (CI).

EtherNet/IP extends EtherNet technology to include the CIP application protocol (**C**ommon **I**ndustrial **P**rotocol). CIP is known in the field of automation engineering because it is also used for DeviceNet and ControlNet as an application protocol.

7.2 Process data exchange

Up to 64 process data words can be exchanged with an EtherNet/IP master (scanner) depending on the use of the UFR41B unit. The EtherNet/IP master (scanner) sets the process data length when opening the connection.

In addition to a controlling "Exclusive Owner Connection", up to two "Listen Only Connections" are available. This means the actual values of the drive can also be read out by stand-by controllers or visualization devices.

If one controlling connection is already active via Modbus/TCP, an "Exclusive Owner Connection" cannot be activated via EtherNet/IP without a power-on reset.

Timeout response

The timeout status is triggered by the UFR41B option. The timeout interval must be set by the EtherNet/IP master (scanner) when the connection is established. The EtherNet/IP specification refers to a "Requested Packet Interval (RPI)" instead of a timeout interval.

The timeout interval displayed in the MOVITOOLS® MotionStudio parameter tree results from the Requested Packet Interval (RPI) multiplied with the "Timeout Multiplier".

This timeout interval is retained in the device when an "Exclusive Owner Connection" is removed, and the device switches to timeout status after the timeout interval has elapsed. The timeout status is displayed on the front of the UFR41B option by the flashing red L13 LED.

A you can only activate the timeout delay via the bus, you must not change the value via MOVITOOLS® MotionStudio.

The timeout state causes the response programmed in the gateway program to be executed.

The timeout state can be reset via EtherNet/IP as follows:

- Via reset service of the identity object (class 0x01, instance 0x01, undetermined attribute)
- By re-establishing the connection
- Via the reset bit in the control word

7.3 CIP object directory

In the Common Industrial Protocol, all unit data can be accessed via objects. The objects listed in the following table are integrated in the UFR41B option.

Class [hex]	Name
01	Identity object
02	Message Router Object
04	Assembly Object
06	Connection Manager Object
07	Register Object
0F	Parameter Object
64	Vardata Object
F5	TCP/IP Interface Object
F6	Ethernet Link Object

The meaning of the objects and a description of how to access them is given in the following section.

Identity object

- The identity object contains general information on the EtherNet/IP device.
- Class code: 01_{hex}

Class

Attribute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0001	Revision 1
2	Get	Max Instance	UINT	0001	Maximum instance

Instance 1

Attribute	Access	Name	Data type	Default value [hex]	Description
1	Get	Vendor ID	UINT	013B	SEW-EURODRIVE GmbH & Co KG
2	Get	Device Type	UINT	0065	Manufacturer-specific type
3	Get	Product Code	UINT	0006	Product no. 6: UFR41B
4	Get	Revision	STRUCT of		Revision of the identity object, depends on firmware version
		Major Revision	USINT		
		Minor Revision	USINT		
5	Get	Status	WORD		See table in "Coding the attribute 5 status"
6	Get	Serial number	UDINT		Unique serial number
7	Get	Product Name	SHORT_ STRING	SEW GATEWAY UFR41B	Product name

- Coding attribute 5 "status":

Bit	Name	Description
0	Owned	Controlling connection is active
1	-	Reserved
2	Configured	Configuration complete
3	-	Reserved
4 - 7	Extended Device Status	See table "Coding the extended device status"
8	Minor Recoverable Fault	Minor fault that can be remedied
9	Minor Unrecoverable Fault	Minor fault that cannot be remedied
10	Major Recoverable Fault	Major fault that cannot be remedied
11	Major Unrecoverable Fault	Major fault that cannot be remedied
12 - 15	-	Reserved

- Coding of the "extended device status " (bits 4 - 7):

Value [binary]	Description
0000	Unknown
0010	At least one faulty IO connection
0011	No IO connection established
0110	At least one IO connection active

Supported services

Service Code [hex]	Service Name	Class	Instance
01	Get_Attributes_All	X	X
05	Reset	-	X
0E	Get_Attribute_Single	X	X

Message router object

- The message router object provides information on the implemented objects.
- Class code: 02_{hex}

Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0001	Revision 1

Instance 1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Object_List	STRUCT of		Object list comprising: • Number of objects • List of objects
		Number	UINT	0009	
		Classes	ARRAY of UINT	01 00 02 00 04 00 06 00 07 00 0F 00 64 00 F5 00 F6 00	
2	Get	Number Available	UINT	0009	Maximum number of connections

Supported services

Service code [hex]	Service Name	Class	Instance
01	Get_Attributes_All	X	-
0E	Get_Attribute_Single	X	X

Assembly object

- The assembly object is used to access the UFR41B process data. IO connections can be created for the instances of the assembly object to exchange cyclic process data.
- Class code: 04_{hex}

Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0002	Revision 2
2	Get	Max Instance	UINT	0082	Maximum instance

Instance 162 - SEW PO data range

This instance is used to access the UFR41B process output data. MOVIDRIVE® can be controlled by only one scanner. Therefore, only one connection can be established with this instance.

Attri-bute	Access	Name	Data type	Default value [hex]	Description
3	Get	Data	Array of BYTE	-	OUTPUT assembly

Instance 121 - "Heartbeat"

This instance is accessed when the scanner wants to establish an input only connection. No process output data is sent with this type of connection. It is used only to read process input data.

Attri-bute	Access	Name	Data type	Default value [hex]	Description
3	Get	Data	Array of BYTE	-	OUTPUT assembly Data size = 0

Instance 172 - SEW PI data range

This instance is used to access the UFR41B process output data. Several multicast connections or a point-to-point connection can be established to this instance.

Attri-bute	Access	Name	Data type	Default value [hex]	Description
3	Get	Data	Array of BYTE	-	INPUT assembly

TIP

The names "INPUT assembly" and "OUTPUT assembly" refer to the processes as seen from the network’s point of view. "INPUT assembly" produces data on the network, an "OUTPUT assembly" consumes data from the network.

Supported services

Service code [hex]	Service Name	Class	Instance 161	Instance 121	Instance 171
0E	Get_Attribute_Single	X	X	-	X

Register object

- The register object is used to access an SEW parameter index.
- Class code: 07_{hex}

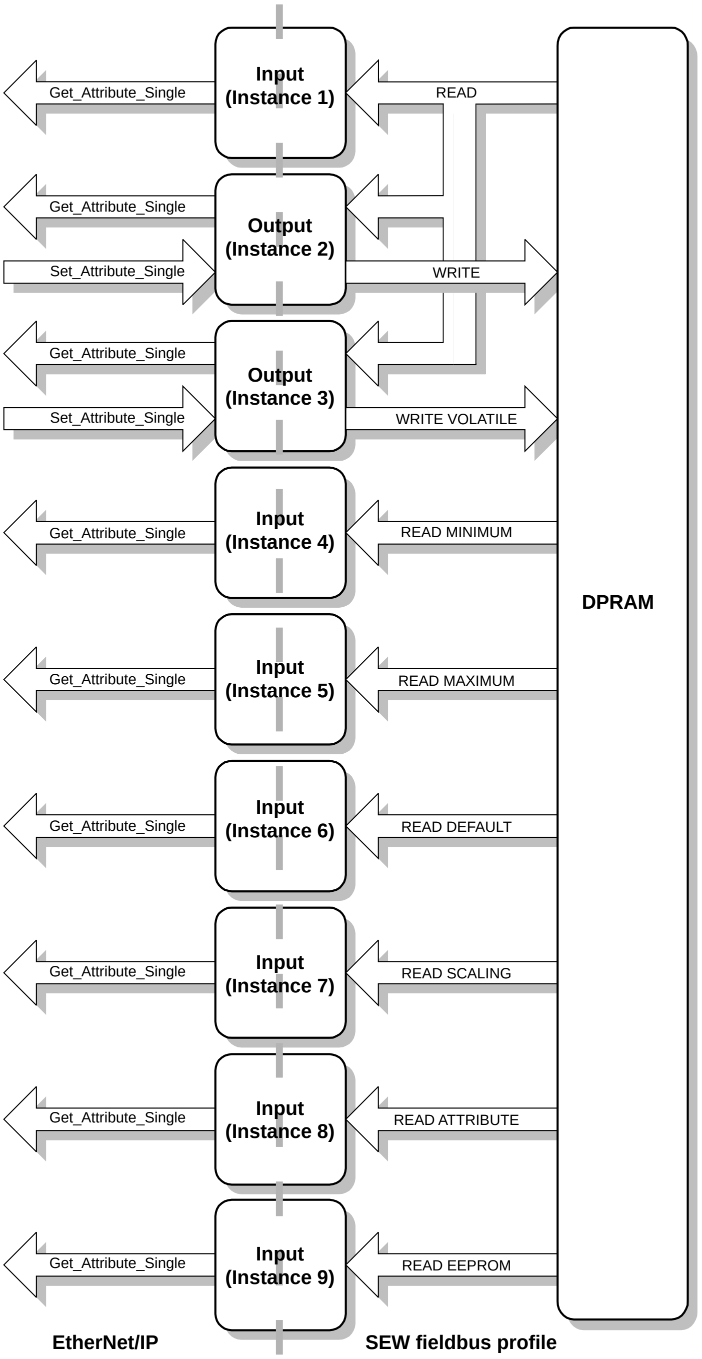
Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
2	Get	Max Instance	UINT	0009	Maximum instance

The MOVILINK® parameter services are mapped in the nine instances of the register object. The "Get_Attribute_Single" and "Set_Attribute_Single" services are used for access.

As the register object is designed so that INPUT objects can only be read and OUTPUT objects can be read and written, the options listed in the following table are available for addressing the parameter channel.

Instance	INPUT/OUTPUT	Resulting MOVILINK® service with	
		Get_Attribute_Single	Set_Attribute_Single
1	INPUT	READ parameter	Invalid
2	OUTPUT	READ	WRITE parameter
3	OUTPUT	READ	WRITE VOLATILE parameter
4	INPUT	READ MINIMUM	Invalid
5	INPUT	READ MAXIMUM	Invalid
6	INPUT	READ DEFAULT	Invalid
7	INPUT	READ SCALING	Invalid
8	INPUT	READ ATTRIBUTE	Invalid
9	INPUT	READ EEPROM	Invalid



54185BEN

Figure 1: Description of the parameter channel

Instance 1 - 9

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Bad Flag	BOOL	00	0 = good / 1 = bad
2	Get	Direction	BOOL	00 01	Input register Output register
3	Get	Size	UINT	0060	Data length in bits (96 bits = 12 bytes)
4	Get/Set	Data	ARRAY of BITS		Data in format of the SEW parameter channel

TIPS

Explanation of the attributes:

- Attribute 1 indicates whether an error occurred during the previous access to the data field.
- Attribute 2 indicates the direction of the instance.
- Attribute 3 indicates the data length in bits
- Attribute 4 represents the parameter data. When accessing attribute 4, the SEW parameter channel must be attached to the service telegram. The SEW parameter channel consists of the elements listed in the following table.

<table>
<tr><th>Name</th><th>Data type</th><th colspan="4">Description</th></tr>
<tr><td>Index</td><td>UINT</td><td colspan="4">SEW unit index</td></tr>
<tr><td>Data</td><td>UDINT</td><td colspan="4">Data (32 bit)</td></tr>
<tr><td>Subindex</td><td>BYTE</td><td colspan="4">SEW unit subindex</td></tr>
<tr><td>Reserved</td><td>BYTE</td><td colspan="4">Reserved (must be '0')</td></tr>
<tr><td>Subaddress 1</td><td>BYTE</td><td>0</td><td rowspan="2">Parameter of the UFR41B itself</td><td>1 ...</td><td>e.g. SBus address of units connected to the SBus of UFR41B</td></tr>
<tr><td>Subchannel 1</td><td>BYTE</td><td>0</td><td>3</td><td>Lower-level bus system, e.g. SBus 1</td></tr>
<tr><td>Subaddress 2</td><td>BYTE</td><td colspan="4">Reserved (must be '0')</td></tr>
<tr><td>Subchannel 2</td><td>BYTE</td><td colspan="4">Reserved (must be '0')</td></tr>
</table>

The subchannels and subaddresses apply depending on the lower-level bus system from MOVI-PLC® *advanced* DHR41B to the drives.

For a schematic representation of the parameter access to lower-level units, refer to the chapter "Appendix".

Sub-channel 1	Interface	Value range subaddress 1
0	UFR41B itself	0
1	Reserved	0
2	EtherCAT X36 (in preparation)	
3	SBus1 (X33 and X26)	1 - 16
4	SBus2 (X32)	17 - 32

Supported services

Service code [hex]	Service Name	Instance
0x0E	Get_Attribute_Single	X
0x10	Set_Attribute_Single	X

Parameter object

- In exceptional cases, you can also use the parameter object to access an SEW parameter channel.
- Class code: 0F$_{hex}$

Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0001	Revision 1
2	Get	Max Instance	UINT	0005	Maximum instance
8	Get	Parameter Class Descriptor	UINT	0009	Bit 0: Supports parameter instances Bit 3: Parameters are saved permanently
9	Get	Configuration Assembly Interface	UINT	0000	Configuration assembly is not supported.

The instances of the parameter object should only be used to access SEW parameters when the EtherNet/IP scanner does not support the option to attach user-defined data to the services "Get_Attribute_Single" and "Set_Attribute_Single."

When you use the parameter object, it takes a number of steps to address a parameter index.

- First, the address of the required parameter is set in instances 1 to 4.
- Next, instance 5 is used to access the parameter that is addressed in instances 1 to 4.

Access to an SEW parameter index via the parameter object is complicated and prone to errors. Consequently, this process should only be used when the EtherNet/IP scanner does not support configuration using the mechanisms of the register object.

Instance 1 - SEW parameter index

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Set	Parameter Value	UINT	207A	Index of the parameter
2	Get	Link Path Size	USINT	00	No link is specified
3	Get	Link Path	Packed EPATH	00	Not used
4	Get	Descriptor	WORD	0000	Read/write parameter
5	Get	Data type	EPATH	00C7	UINT
6	Get	Data Size	USINT	02	Data length in bytes

Instance 2 - SEW subindex

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Set	Parameter Value	UINT	0000	Low byte contains the subindex
2	Get	Link Path Size	USINT	00	No link is specified
3	Get	Link Path	Packed EPATH	00	Not used
4	Get	Descriptor	WORD	0000	Read/write parameter
5	Get	Data type	EPATH	00C7	UINT
6	Get	Data Size	USINT	02	Data length in bytes

Instance 3 - SEW subparameter 1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Set	Parameter Value	UINT	0000	Low byte contains subaddress 1 High byte contains subchannel 1
2	Get	Link Path Size	USINT	00	No link is specified
3	Get	Link Path	Packed EPATH	00	Not used
4	Get	Descriptor	WORD	0000	Read/write parameter
5	Get	Data type	EPATH	00C7	UINT
6	Get	Data Size	USINT	02	Data length in bytes

Instance 4 - SEW subparameter 2

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Set	Parameter Value	UINT	0000	Low byte contains subaddress 2 High byte contains subchannel 2
2	Get	Link Path Size	USINT	00	No link is specified
3	Get	Link Path	Packed EPATH	00	Not used
4	Get	Descriptor	WORD	0000	Read/write parameter
5	Get	Data type	EPATH	00C7	UINT
6	Get	Data Size	USINT	02	Data length in bytes

Instance 5 - SEW read/write

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Set	Parameter Value	UDINT		The set service executes write access to the parameters addressed in instances 1 to 4. The get service executes read access to the parameters addressed in instances 1 to 4.
2	Get	Link Path Size	USINT	00	No link is specified
3	Get	Link Path	Packed EPATH	00	Not used
4	Get	Descriptor	WORD	0000	Read/write parameter
5	Get	Data type	EPATH	00C8	UDINT
6	Get	Data Size	USINT	04	Data length in bytes

Supported services

Service code [hex]	Service Name	Class	Instance
0E	Get_Attribute_Single	X	X
10	Set_Attribute_Single	-	X

Vardata object

- This manufacturer-specific object is required to use the engineering option of some of the software tools provided by SEW-EURODRIVE.
- Class code: 64_{hex}

Class

None of the class attributes are supported.

Instance 1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Data	ARRAY OF SINT	-	-
2	Get	Size	UINT	00F2	Maximum data length in bytes

Supported services

Service code [hex]	Service Name	Instance attribute 1	Instance attribute 2
0E	Get_Attribute_Single	X	X
32	Vardata (custom)	X	-

The standardized service "Get_Attribute_Single" (Service Code 0x0E) returns a data stream with the maximum data length (attribute 2) when instance attribute 1 is accessed. The data content is filled with zeros. If a data stream is added to the request telegram (Service Type Custom), this data is returned in a mirrored form (Vardata test mode).

The Vardata service (service code 0x32) is a manufacturer-specific service. With this service, the telegram structure for the request and response is the same. The telegram contains routing information, the data length of the Vardata user data telegram and the actual Vardata layer 7 telegram. The data length of the Vardata layer 7 telegram is variable.

The following table shows the complete telegram structure.

Name	Data type
Subaddress 1	BYTE
Subchannel 1	BYTE
Subaddress 2	BYTE
Subchannel 2	BYTE
Data Len Low	BYTE
Data Len High	BYTE
Reserved	BYTE
Reserved	BYTE
FC	BYTE
Vardata	Array of BYTE

TCP/IP interface object

- The TCP/IP interface object enables the IP parameters to be configured via EtherNet/IP.
- Class code: F5$_{hex}$

Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0001	Revision 1
2	Get	Max Instance	UINT	0001	Maximum instance
3	Get	Number of Instances	UINT	0001	DHR41B has one TCP/IP interface

Instance 1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Status	DWORD	00000001	Valid configuration
2	Get	Configuration Capability	DWORD	00000014	The interface configuration attribute (5) is writable. The DHCP can be used for configuration.
3	Set	Configuration Control	DWORD	00000002	0 = The unit uses the stored IP parameters at startup. 2 = The unit waits for its IP configuration via DHCP at startup.
4	Get	Physical Link Object	STRUCT of		Reference to the EtherNet link object (class code 0xF6) as sublayer.
		Path Size	UINT	0002	
		Path	Padded EPATH	20 F6 24 01	
5	Set	Interface Con-figuration	STRUCT of		
		IP Address	UDINT		Current IP address
		Network Mask	UDINT		Current subnetwork mask
		Gateway Address	UDINT		Current standard gateway
		Name Server	UDINT	00000000	DNS is not supported
		Name Server 2	UDINT	00000000	DNS is not supported
		Domain Name	STRING	sew.de	
6	Get	Host Name	STRING		Not used

Supported services

Service code [hex]	Service Name	Class	Instance
01	Get_Attributes_All	X	_
0E	Get_Attribute_Single	X	X
10	Set_Attribute_Single	-	X

EtherNet link object

- Information on the Ethernet communication interface is stored in the Ethernet link object.
- Class code: F6$_{hex}$

Class

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0002	Revision 2
2	Get	Max Instance	UINT	0002	Maximum instance
3	Get	Number of Instances	UINT	0002	DHR41B has two Ethernet interfaces

Instance 1 Ethernet connection X30:1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Interface Speed	UDINT	00000064	Default value = 100 → Transmission speed in MBit/s
2	Get	Interface Flags	DWORD		• Bit 0 displays the active link • Bit 1 displays full duplex mode • Bit 2 ... bit 4 signal negotiation status • Bit 5 shows whether the manual setting has to be reset • Bit 6 indicates a local hardware fault
3	Get	Physical Address	ARRAY of 6 USINTs	00 0F 69 xx xx xx	MAC ID SEW MAC OUI: 00 0F 69

Instance 2 Ethernet connection X30:2

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Interface Speed	UDINT	00000064	Default value = 100 → transmission speed in MBit/s
2	Get	Interface Flags	DWORD		• Bit 0 displays the active link • Bit 1 displays full duplex mode • Bits 2 - bit 4 indicate the negotiation status • Bit 5 shows whether the manual setting has to be reset • Bit 6 indicates a local hardware fault
3	Get	Physical Address	ARRAY of 6 USINTs	00 0F 69 xx xx xx	MAC ID SEW MAC OUI: 00 0F 69

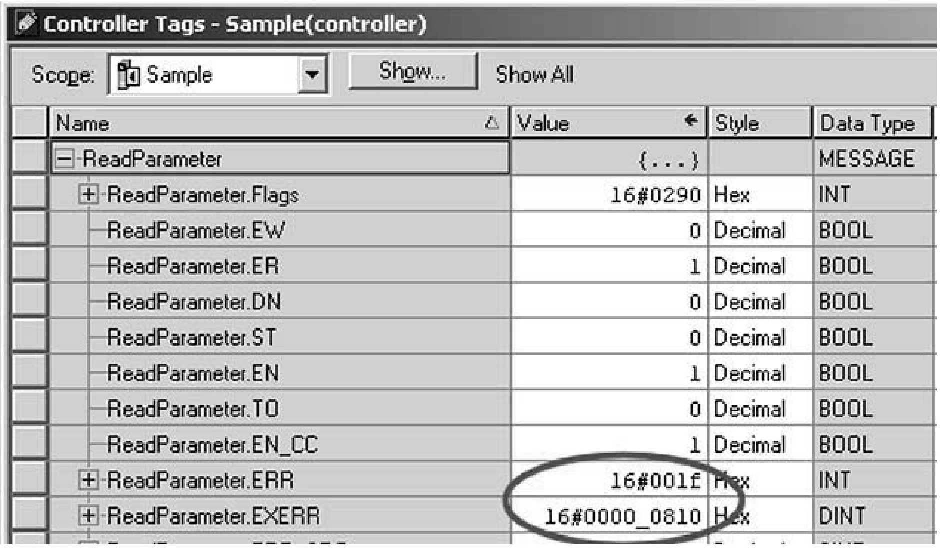
Supported services

Service code [hex]	Service Name	Class	Instance
01	Get_Attributes_All	X	_
0E	Get_Attribute_Single	X	X

7.4 Return codes of the parameterization via explicit messages

If a parameter request via explicit messages fails, a fault code can be used to determine the cause. An error can be generated either by the UFR41B option, by the EtherNet/IP system, or by a timeout.

The general error code (ERR) and the additional code (EXERR) can be read out from the status registers of the message tags (see figure below).



Controller Tags - Sample(controller)

Scope: Sample   Show...   Show All

Name	Value	Style	Data Type
⊟ ReadParameter	{...}		MESSAGE
⊞ ReadParameter.Flags	16#0290	Hex	INT
ReadParameter.EW	0	Decimal	BOOL
ReadParameter.ER	1	Decimal	BOOL
ReadParameter.DN	0	Decimal	BOOL
ReadParameter.ST	0	Decimal	BOOL
ReadParameter.EN	1	Decimal	BOOL
ReadParameter.TO	0	Decimal	BOOL
ReadParameter.EN_CC	1	Decimal	BOOL
⊞ ReadParameter.ERR	16#001f	Hex	INT
⊞ ReadParameter.EXERR	16#0000_0810	Hex	DINT

11937AXX

Return codes of EtherNet/IP

EtherNet/IP-specific return codes are returned in the error telegram if the data format is not maintained during the transfer or if a service is performed that has not been implemented. The coding of these return codes is described in the EtherNet/IP specification (see section "General error codes"). The General Error Code of a manufacturer-specific return code is $1F_{hex}$.

SEW-specific return codes

The return codes that the UFR41B option or lower-level units send in the event of incorrect parameterization are described in section "MOVILINK®-specific return codes". In conjunction with EtherNet/IP, the return codes are returned in the following format. The following table shows the data format for a parameter response telegram.

	Byte offset			
	0	**1**	**2**	**3**
Function	General error code	Additional code Length (words)	Additional Code Word 1 (lowbyte)	Additional Code Word 1 (highbyte)
Example	$1F_{hex}$ Vendor-specific	01_{hex} only low word (word 1)	10_{hex} MOVILINK® Additional Error Code	08_{hex} MOVILINK® Error Class

In the example above, MOVILINK® error class 08 (General Error) is shown in the high byte of the additional code. The MOVILINK® additional error code 10 (invalid index) is shown in the low byte of the additional code. This information shows that the system tried to access a unit index that does not exist.

Timeout of explicit messages

The timeout is triggered by the UFR41B option. The timeout interval must be set by the master after the connection has been established. The EtherNet/IP specification refers to an "Expected packet rate" rather than a timeout time in this case. The expected packet rate is calculated from the timeout delay as follows:

$t_{Timeout\_ExplicitMessages} = 4 \times t_{Expected\_Packet\_Rate\_ExplicitMessages}$

It can be set using connection object class 5, instance 1, attribute 9. The range of values runs from 0 ms to 655535 ms in 5 ms steps.

If there is a timeout for the explicit messages, this connection type is automatically disconnected for the explicit messages. This is the default setting for EtherNet/IP. The connection for these explicit messages must be re-established to communicate with these messages again. The timeout is **not** forwarded to the gateway program.

General error codes

General error code (hex)	Error name	Description
00	Success	Successful
01	Connection failure	A connection-specific service has failed.
02	Resource unavailable	The source required for performing the service is unavailable.
03		Reserved
04	Path segment error	The processing node cannot interpret the "Path segment identifier" or the segment syntax.
05	Path destination unknown	The "Path" refers to an object class, object instance or a structural element that is not supported by the processing node.
06 - 07		Reserved
08	Service not supported	The service is not supported for the selected class/instance.
09	Invalid attribute value	Invalid attribute data have been sent.
0A - 0B		
0C	Object state conflict	The selected object cannot perform the service in its current status.
0D		Reserved
0E	Attribute not settable	It is not possible to access the selected object for writing.
10	Device state conflict	The current status of the device makes it impossible to perform the required service.
11 - 12		Reserved
13	Not enough data	The length of the transferred data is too short for the service to be performed.
14	Attribute not supported	The selected attribute is not supported.
15	Too much data	The length of the transferred data is too long for the service to be performed.
16	Object does not exist	The selected object is not implemented in the device.
17-1D		Reserved
1E	Embedded Service Error	Internal processing error
1F	Vendor specific error	Manufacturer-specific error (see "Fieldbus Unit Profile" manual).
20	Invalid parameter	Invalid parameter. This error message is used when a parameter does not satisfy the requirements of the specification and/or the requirements of the application.
21 - FF		Reserved

MOVILINK®-specific return codes

The following table shows the MOVILINK®-specific return codes (MOVILINK® "Error Class" and "Additional Code") in the event of an incorrect parameterization.

MOVILINK®		
Error Class	**Additional Code**	**Description**
0x05	0x00	Unknown error
	0x01	Illegal Service
	0x02	No Response
	0x03	Different Address
	0x04	Different Type
	0x05	Different Index
	0x06	Different Service
	0x07	Different Channel
	0x08	Different Block
	0x09	No Scope Data
	0x0A	Illegal Length
	0x0B	Illegal Address
	0x0C	Illegal Pointer
	0x0D	Not enough memory
	0x0E	System Error
	0x0F	Communication does not exist
	0x10	Communication not initialized
	0x11	Mouse conflict
	0x12	Illegal Bus
	0x13	FCS Error
	0x14	PB Init
	0x15	SBUS - Illegal Fragment Count
	0x16	SBUS - Illegal Fragment Type
	0x17	Access denied
		Not used

MOVILINK®		
Error Class	**Additional Code**	**Description**
0x08	0x00	No Error
	0x10	Illegal Index
	0x11	Not yet implemented
	0x12	Read only
	0x13	Parameter Blocking
	0x14	Setup runs
	0x15	Value too large
	0x16	Value too small
	0x17	Required Hardware does not exist
	0x18	Internal Error
	0x19	Access only via RS485 (via X13)
	0x1A	Access only via RS485 (via XT)
	0x1B	Parameter protected
	0x1C	"Controller inhibit" required
	0x1D	Value invalid
	0x1E	Setup started
	0x1F	Buffer overflow
	0x20	"No Enable" required
	0x21	End of File
	0x22	Communication Order
	0x23	"IPOS Stop" required
	0x24	Autosetup
	0x25	Encoder Nameplate Error
	0x29	PLC State Error

8 Configuration and Startup (Modbus/TCP)

This section provides information about the configuration of the Modbus/TCP master and startup of the inverter for fieldbus operation. Prerequisite is the correct connection and setting of the IP address parameters of UFR41B in accordance with chapter "Assembly and Installation Instructions".

8.1 Unit description file for Modbus/TCP

TIP

There are no specific unit description files for Modbus/TCP.

8.2 Configuring the master (Modbus scanner)

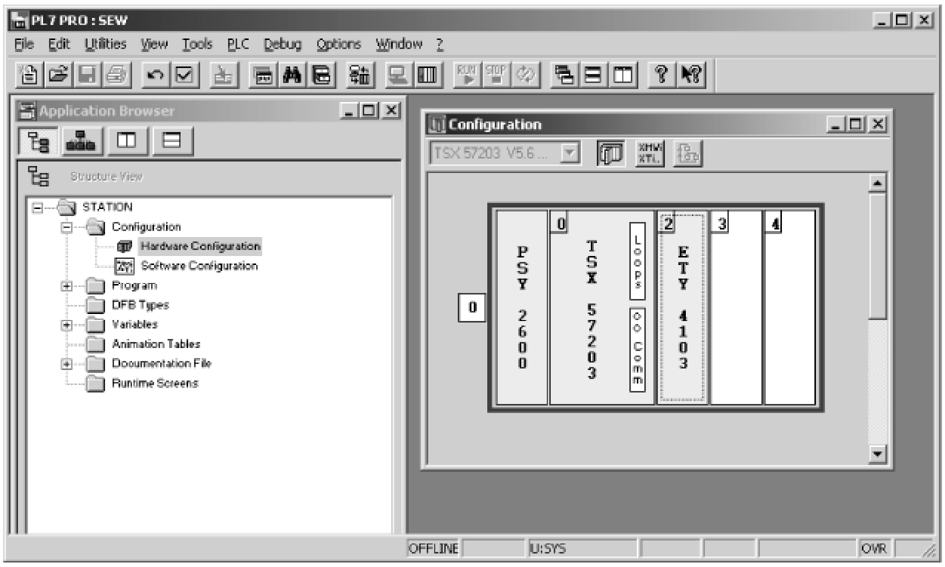
The first example refers to the configuration and programming of a Schneider Electric control system TSX Premium P57203 using the programming software PL7 PRO. An ETY4103 is used as the Ethernet component. The information and illustrations are based on the English version of the PL7 PRO software.

TIP

- Enter values in PL7 PRO using the keypad.
- As the Ethernet, use bus master components from Schneider Electric that support IO scanning. The Modbus/TCP interface module for SEW drives cannot be addressed via "Peer COP". However, Ethernet bus masters that only support "Peer COP" can access the drives from the PLC program using read and write commands.

Hardware configuration (control structure)

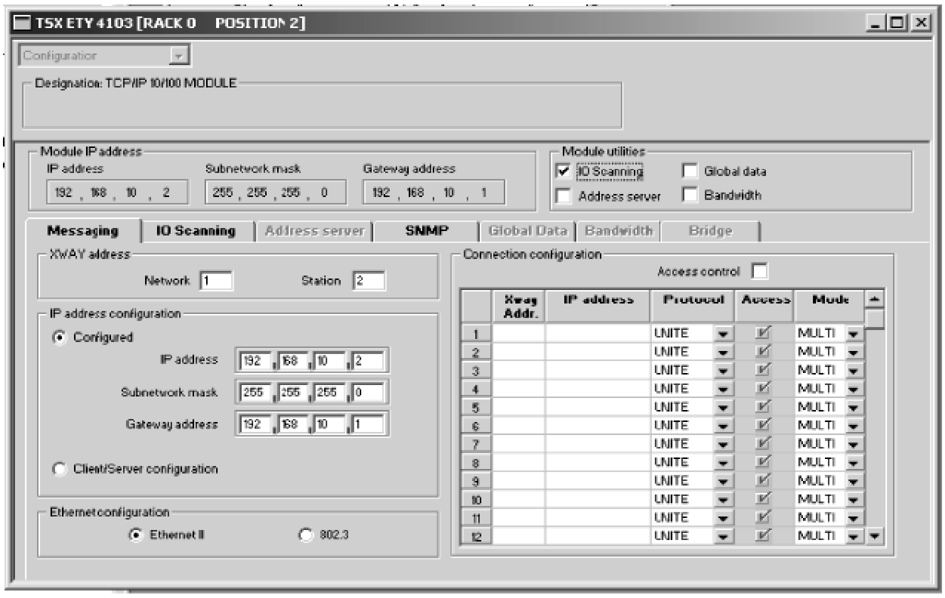
- Start PL7 PRO and enter the control type.
- Enter the hardware configuration for the control system in the application browser under STATION / Configuration / Hardware Configuration .



10815AXX

Settings for the Ethernet component

- To open the configuration window, double-click on the Ethernet component.
- If you have a non-extendable rack, enter a "1" in the "Network" input field in the "XWAY address" section.
- Enter the number of the slot that the Ethernet component is plugged into (here: 2) in the input field "Station" in the "XWAY address" section. 2). In this case, the XWAY address is 1.2.
- In the section "IP address configuration" select the radio button "Configured". Enter the IP address and the network parameters in the input fields "IP address", "Subnetwork mask" and "Gateway address". If the control system is to receive the address parameters via a DHCP server, select the radio button "Client/Server configuration" in the section "IP address configuration".
- In the "Ethernet configuration" section, select the radio button "Ethernet II".
- In the "Module utilities" section, select the checkbox "IO Scanning".


TSX ETY 4103 [RACK 0 POSITION 2]

Configuration

Designation: TCP/IP 10/100 MODULE

Module IP address
IP address: 192 . 168 . 10 . 2
Subnetwork mask: 255 . 255 . 255 . 0
Gateway address: 192 . 168 . 10 . 1

Module utilities
☑ IO Scanning ☐ Global data
☐ Address server ☐ Bandwidth

XWAY address
Network: 1 Station: 2

IP address configuration
● Configured
IP address: 192 . 168 . 10 . 2
Subnetwork mask: 255 . 255 . 255 . 0
Gateway address: 192 . 168 . 10 . 1
○ Client/Server configuration

Ethernet configuration
● Ethernet II ○ 802.3

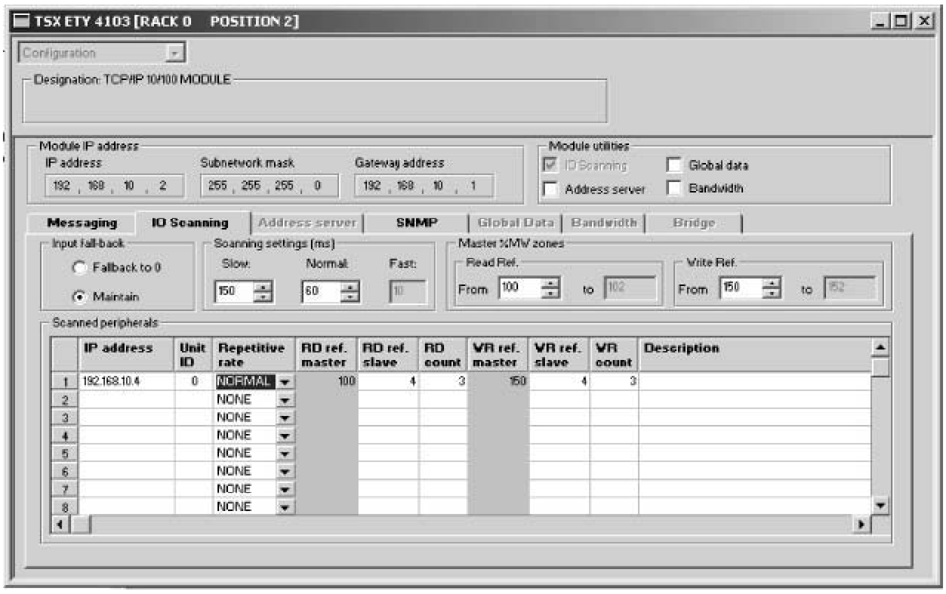
Connection configuration
Access control ☐

	Xway Addr.	IP address	Protocol	Access	Mode
1			UNITE	✓	MULTI
2			UNITE	✓	MULTI
3			UNITE	✓	MULTI
4			UNITE	✓	MULTI
5			UNITE	✓	MULTI
6			UNITE	✓	MULTI
7			UNITE	✓	MULTI
8			UNITE	✓	MULTI
9			UNITE	✓	MULTI
10			UNITE	✓	MULTI
11			UNITE	✓	MULTI
12			UNITE	✓	MULTI

10816AXX

Addressing the drive using IO scanning

- Choose the "IO Scanning" tab page. In this tab page you specify which of the stations connected to the Modbus are to exchange cyclical data.
- In the section "Master %MW zones" enter the control memory areas that are to be used to exchange cyclical data with the Modbus stations. You will use the memory addresses later in your PLC program.
- Enter the following in the "Scanned peripherals" group:
  – In the "IP address" input field, enter the IP address of the SEW drive.
  – In the "Unit ID" input field, enter the value "0".
  – In the "Repetitive rate" dropdown menu, enter the cycle time that is used to address the stations.
  – Enter the value "4" in the input fields "RD ref.slave" and "WR ref. slave" as the cyclical process data are available from offset 4.
  – In the input fields "RD count" and "WR count" enter the number of words to be exchanged. The values must be the same in both fields. For the UFR41B option, you can enter between 1 and 64 words.



Designation: TCP/IP 10/100 MODULE

Module IP address
IP address: 192 . 168 . 10 . 2
Subnetwork mask: 255 . 255 . 255 . 0
Gateway address: 192 . 168 . 10 . 1

Module utilities
☑ IO Scanning ☐ Global data
☐ Address server ☐ Bandwidth

Input fall-back
○ Fallback to 0
● Maintain

Scanning settings (ms)
Slow: 150 Normal: 60 Fast: 10

Master %MW zones
Read Ref. From 100 to 102
Write Ref. From 150 to 152

Scanned peripherals

	IP address	Unit ID	Repetitive rate	RD ref. master	RD ref. slave	RD count	WR ref. master	WR ref. slave	WR count	Description
1	192.168.10.4	0	NORMAL	100	4	3	150	4	3	
2			NONE							
3			NONE							
4			NONE							
5			NONE							
6			NONE							
7			NONE							
8			NONE							

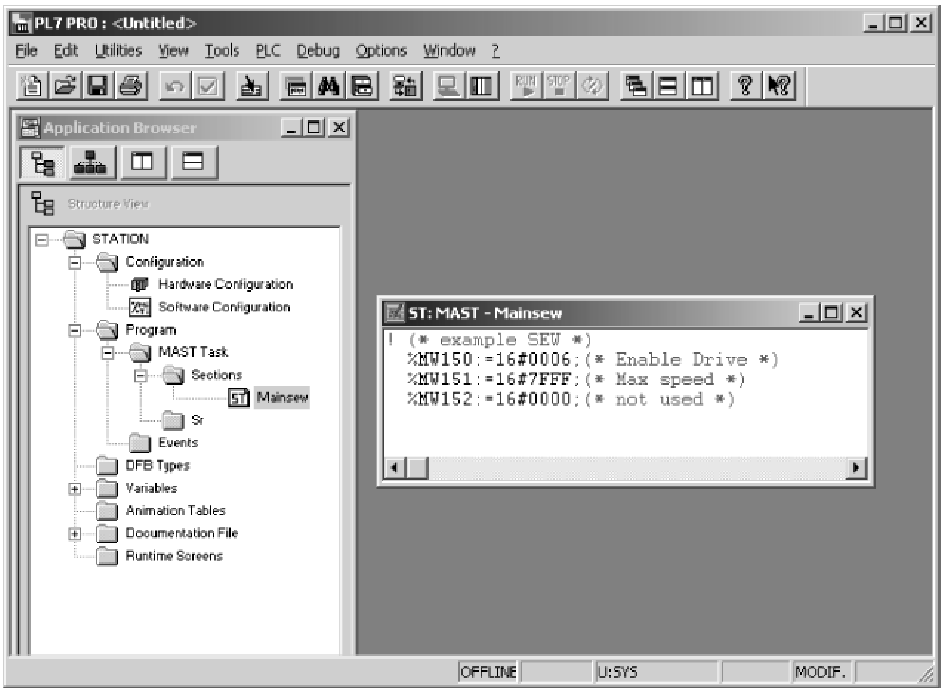
10817AXX

- Click on the "Confirm" button to confirm the rack configuration and the global configuration.
- Once you have transferred your settings and started the program, the color of LED L13 (NETWORK/STATUS) of UFR41B changes to green (see section "Status LEDs of the UFR41B option").

8.3 Project planning examples in PL7 PRO

8.3.1 UFR41B fieldbus gateway with 16 PD data exchange

1. Set the IP address of the DFE33B option see chapter "Setting the IP address parameters").
2. Integrate the UFR41B fieldbus gateway into the configuration for the I/O scanning according to section "Configuring the master (Modbus scanner)".
3. Now, the integration into the PLC project can be performed.
4. Create a new section in PL7 PRO in the application browser under [Station] / [Program] / [Mast Task] / [Sections] .
5. In this example, the setpoints for the drive start from MW150 (see following figure).



```
! (* example SEW *)
  %MW150:=16#0006;(* Enable Drive *)
  %MW151:=16#7FFF;(* Max speed *)
  %MW152:=16#0000;(* not used *)
```

10818AXX

6. Now save the project and transfer it to the PLC. The PLC is set to RUN mode.

   You can now read the actual values from the UFR41B fieldbus gateway and write setpoint values.

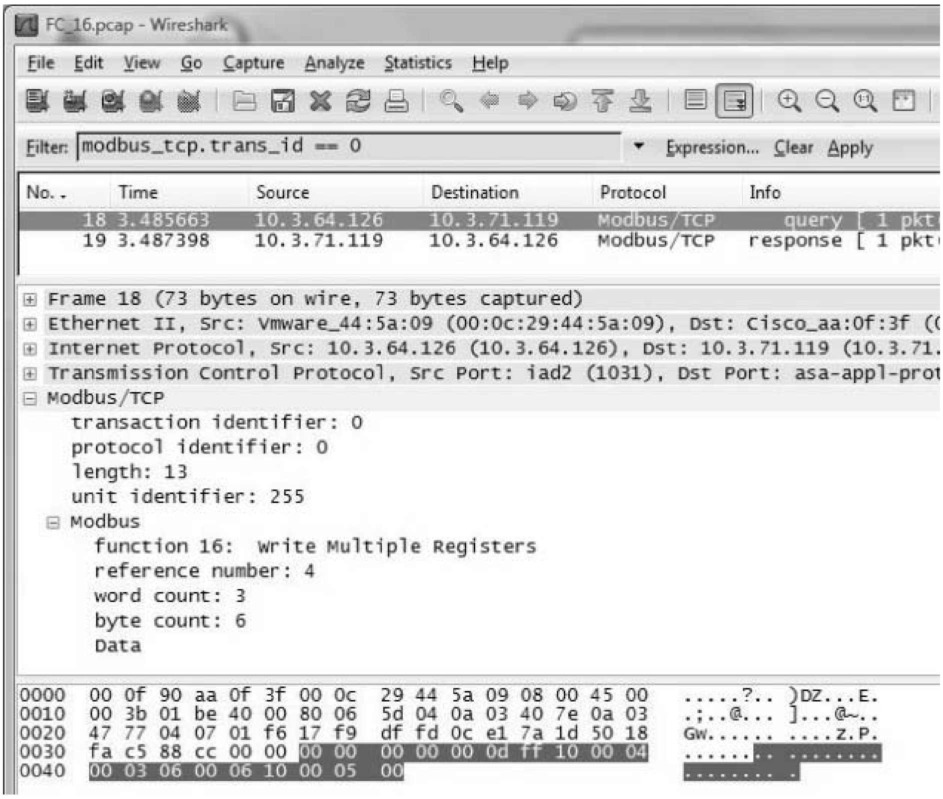
   The process data should correspond with the values displayed in the gateway program of MOVITOOLS® MotionStudio (see following figure).

8.4 Examples for data exchange via Modbus/TCP

As there are many different master systems and software solutions available for Modbus/TCP for standard PCs, there is no "reference controller" the examples are based on. This is why this section gives detailed examples for the telegram structure.

You can compare the telegram structure in your own applications with the telegram structure in these examples for troubleshooting. There are simple tools for recording telegrams via the Ethernet network, e.g. Wireshark (see following figure), Packetizer etc. These freeware tools are available on the Internet.

Observe that tracing all Ethernet telegrams in a network is only possible with a tab, hub or a switch with a port mirror function. The telegrams sent from and to the PC that is also used for recording, can always be written of course.



FC_16.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: modbus_tcp.trans_id == 0
Expression... Clear Apply

```
No. .  Time        Source        Destination   Protocol    Info
    18 3.485663    10.3.64.126   10.3.71.119   Modbus/TCP     query [ 1 pkt(
    19 3.487398    10.3.71.119   10.3.64.126   Modbus/TCP  response [ 1 pkt(

⊞ Frame 18 (73 bytes on wire, 73 bytes captured)
⊞ Ethernet II, Src: Vmware_44:5a:09 (00:0c:29:44:5a:09), Dst: Cisco_aa:0f:3f ((
⊞ Internet Protocol, Src: 10.3.64.126 (10.3.64.126), Dst: 10.3.71.119 (10.3.71.
⊞ Transmission Control Protocol, Src Port: iad2 (1031), Dst Port: asa-appl-prot
⊟ Modbus/TCP
    transaction identifier: 0
    protocol identifier: 0
    length: 13
    unit identifier: 255
  ⊟ Modbus
      function 16:  Write Multiple Registers
      reference number: 4
      word count: 3
      byte count: 6
      Data

0000  00 0f 90 aa 0f 3f 00 0c  29 44 5a 09 08 00 45 00   .....?.. )DZ...E.
0010  00 3b 01 be 40 00 80 06  5d 04 0a 03 40 7e 0a 03   .;..@... ]...@~..
0020  47 77 04 07 01 f6 17 f9  df fd 0c e1 7a 1d 50 18   Gw...... ....z.P.
0030  fa c5 88 cc 00 00 00 00  00 00 00 0d ff 10 00 04   ........ ........
0040  00 03 06 00 06 10 00 05  00                        ........ .
```

12047AXX

The figure above provides an example of how setpoints are sent to the Modbus/TCP slave with IP address 10.3.71.119. The 3 process data words are located from offset 4 (reference number) and are addressed via unit ID 255.

All the other examples merely describe the Modbus/TCP part of the telegram. The TCP/IP part of the telegram, as well as establishing and dropping a TCP/IP connection are not explained in detail.

8.4.1 Writing and reading process data

The process data exchange can be performed either via FC3 (read) and FC16 (write), or FC23 (read and write):

For writing 3 process data words (setpoints) to a Modbus/TCP slave via FC16, the TCP/IP telegram to port 502 is structured as illustrated above.

Byte	Value	Meaning	Interpretation	Help
0	0x00	Transaction identifier		For a detailed description, refer to Modbus/TCP specification and section "Modbus protocol (Modbus/TCP)"
1				
2	0x00	Protocol identifier		
3				
4	0x00	Length field	Number of bytes after byte 5: 3 (number PD) × 2 + 7 = 13	
5	0x0d			
6	0xFF	Unit identifier	Must be 0 or 255	
7	ox10	Function code	Service = FC16 (Write Register)	
8	0x00	Write reference number	Offset where PD start: Must always be 4	
9	0x04			
10	0x00	Write Word Count	Number of PD (here 3): Must be PD 1 - 64	
11	0x03			
12	0x06	Write Byte Count	Number of PD × 2 = 6	
13	0x00	Data	Process output data word 1	Data mapping and definition, see IEC program
14	0x11			
15	0x22		Process output data word 2	
16	0x33			
17	0x44		Process output data word 3	
18	0x55			

Only bytes 0-11 are returned in the response telegram of port 502 of the Modbus/TCP slave, where all values remain unchanged with the exception of byte 5. Byte 5 (low byte length field) is corrected to value 6.

During process data exchange via FC23, the telegram that is used to write and read 3 process data words (PD) each is structured as follows.

Byte	Value	Meaning	Interpretation	Help
0	0x00	Transaction identifier		For a detailed description, refer to Modbus/TCP specification and section 'Modbus protocol (Modbus/TCP)'
1				
2	0x00	Protocol identifier		
3				
4	0x00	Length field	Number of bytes after byte 5: 3 (number PD) × 2 + 11 = 17	
5	0x11			
6	0xFF	Unit identifier	Must be 0 or 255	
7	0x10	Function code	Service = FC23 (Read + Write Register)	
8	0x00	Read reference number	Offset where PD start: Must always be 4	
9	0x04			
10	0x00	Read Word Count	Number of PD (here 3): Must be PD 1 - 64	
11	0x03			
12	0x00	Write reference number	Offset where PD start: Must always be 4	
13	0x04			
14	0x00	Write Word Count	Number of PD (here 3): see read word count	
15	0x03			
16	0x06	Write Byte Count	Number of PD × 2 = 6	
17	0x00	Data	Process output data word 1	Data mapping and definition, see IEC program
18	0x11			
19	0x22		Process output data word 2	
20	0x33			
21	0x44		Process output data word 3	
22	0x55			

The following data bytes are returned in the response telegram of the Modbus/TCP slave:

Byte	Value	Meaning	Interpretation	Help
0	0x00	Transaction identifier		For a detailed description, refer to Modbus/TCP specification and section "Modbus protocol (Modbus/TCP)"
1				
2	0x00	Protocol identifier		
3				
4	0x00	Length field	Number of bytes after byte 5: 3 (number PD) × 2 + 3 = 9	
5	0x09			
6	0xFF	Unit identifier	Must be 0 or 255	
7	0x17	Function code	Service = FC23 (Read + Write Register)	
8	0x06	Write Byte Count	Number of PD × 2 = 6	
9	0x00	Data	Process input data word 1	Data mapping and definition, see IEC program
10	0xAA			
11	0xBB		Process input data word 2	
12	0xCC			
13	0xDD		Process input data word 3	
14	0xEE			

8.4.2 Parameter access

FC23 is suitable for the parameter access via the MOVILINK® parameter channel as it is possible to realize the request to the MOVILINK® service and the collection of the response in one Modbus/TCP service.

To read a parameter, the TCP/IP telegram is structured as follows.

Byte	Value	Meaning	Interpretation	Help
0	0x00	Transaction identifier		For a detailed description, refer to Modbus/TCP specification and section "Modbus protocol (Modbus/TCP)"
1				
2	0x00	Protocol identifier		
3				
4	0x00	Length field	Number of bytes after byte 5: Must be 19 for MOVILINK®	
5	0x13			
6	0x00	Unit identifier	1)	
7	0x17	Function code	Service = FC23 (Read + Write Register)	
8	0x02	Read reference number	Offset where the MOVILINK® parameter channel starts: Must always be 512	
9	0x00			
10	0x00	Read Word Count	Must always be 4 for the MOVILINK® parameter channel.	
11	0x04			
12	0x02	Write reference number	Offset where the MOVILINK® parameter channel starts: Must always be 512	
13	0x00			
14	0x00	Write Word Count	Must always be 4 for the MOVILINK® parameter channel.	
15	0x04			
16	0x08	Write Byte Count	8 bytes MOVILINK®	
17	0x31	Data: MOVILINK® parameter channel	Management byte: 0x31 = read	Data mapping and definition, see IEC program and SEW unit profile
18	0x00		Parameter subindex	
19	0x20		Parameter index: 0x206c = 8300 = Firmware part number	
20	0x6C			
21	0x00		Parameter value Irrelevant for read service	
22	0x00			
23	0x00			
24	0x00			

1) The unit identifier 0 and 0xFF is used to access the parameters of DHR41B directly. For other values, the request is passed on to a lower-level unit. The assignment of the unit identifier to the downstream units on the system buses is determined via the routing table of the DHR41B control configuration. This allows parameter access for inverters that are connected via a DHR41B unit without any restrictions. See the "Appendix" for a schematic representation of parameter access to lower-level units.

The response telegram receives the response to the MOVILINK® read service.

Byte	Value	Meaning	Interpretation	Help
0	0x00	Transaction identifier		For a detailed description, refer to Modbus/TCP specification and section "Modbus protocol (Modbus/TCP)"
1				
2	0x00	Protocol identifier		
3				
4	0x00	Length field	Number of bytes after byte 5: Must be 11 for MOVILINK®	
5	0x11			
6	0x00	Unit identifier	[1]	
7	0x17	Function code	Service = FC23 (Read + Write Register)	
8	0x02	Read reference number	8 bytes MOVILINK®	
17	0x31	Data: MOVILINK® parameter channel	Management byte: 0x31 = read	For data mapping and definition, see unit setting and SEW unit profile
18	0x00		Parameter subindex	
19	0x20		Parameter index: 0x206c = 8300 = Firmware part number	
20	0x6C			
21	0x00		The parameter value 0xA82e5b0d corresponds to the firmware part number 28216102.53	
22	0x00			
23	0x00			
24	0x00			

[1] The unit identifier 0 and 0xFF is used to access the parameters of DHR41B directly. For other values, the request is passed on to a lower-level unit. The assignment of the unit identifier to the downstream units on the system buses is determined via the routing table of the DHR41B control configuration. This allows parameter access for inverters that are connected via a DHR41B unit without any restrictions. See the "Appendix" for a schematic representation of parameter access to lower-level units.

9 Modbus Protocol (Modbus/TCP)

9.1 Introduction

Modbus/TCP is an open protocol based on TCP/IP. It was one of the first protocol types to become standard in industrial Ethernet interfaces for process data transfer.

Modbus frames are exchanged via the TCP/IP port 502. Every master IP address is accepted. Modbus exclusively uses the coding "BIG ENDIAN" (Motorola data format or high byte first).

Access via "Peer COP" is not possible. Make sure that the bus master supports "IO scanning".

9.1.1 Mapping and addressing

The logic Modbus address range is 64 k words and is addressed via the reference number (offset). Four different tables can be in the address range:

- Binary inputs (RO)
- Binary outputs (RW)
- Input register (RO)
- Output register (RW)

The tables can be separate or overlapping.

The UFR41B option provides the following data areas:

- For process data transfer, there is a table that allows for write access (for setpoint values) as well as read access (for actual values).

  This table starts at offset 4 and ends at offset 0FF$_{hex}$. It contains the 1 64 cyclically transferred process data words.
- The process data output words from the controller are also saved in another table. It allows for one or several additional clients (e.g. visualization) to read the current setpoint values.

  This table starts at offset 104$_{hex}$ and ends at offset 1FF$_{hex}$.
- A third table is provided for parameter access.

  This table starts at offset 200$_{hex}$, ends at offset 2FF$_{hex}$ and contains 4 words of the MOVILINK$^{®}$ parameter channel (see "Fieldbus Unit Profile" manual).
- The remaining address scope from offset 400$_{hex}$ to FFFF$_{hex}$ is reserved and must not be addressed.

  The data word at offset 219$_{hex}$ (8606$_{dec}$) is a special case, it allows for writing (and reading) the timeout monitoring time.

TIP

For Schneider Electric control systems:

The address range often starts with 40001$_{hex}$, which corresponds to an offset with the value "0".

9.1.2 Services (function codes)

For process data exchange, parameter data exchange and unit identification, the DHR41B option provides 4 FC.. services (**F**unction **C**odes).

- FC 3 Read Holding Registers
- FC16 Write Multiple Registers
- FC23 Read/Write Multiple Registers
- FC43 Read Device Identification

The FC3 and FC16 services allow reading or writing of one or more registers. FC23 allows a register block to be read and written simultaneously. FC43 can be used for a unit identification by reading out the identity parameters.

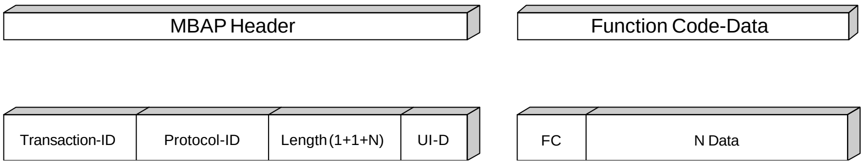
9.1.3 Access

The implemented registers and possible services (function codes) for data exchange are summarized in the following table.

	Meaning when			
Offset (hex)	**Read**	**Write**	**Access**	**Comment**
0 - 3	-	-	-	Reserved
4 - FF	Process input data (actual values)	Process output data (setpoint values)	FC3, FC16, FC23	0 - 64 words
100 - 103	-	-	-	Reserved
104 - 1FF	Process output data (setpoint values)	-	FC3	For reading the setpoint values by a client other than the controlling one
200 - 2FF	Result acyclical parameter channel	Request acyclical parameter channel	FC3, FC16, FC23	4 words
300 - FFFF	-	-	-	Reserved
Special case: 219E (8606_{dec})	Fieldbus timeout interval, read value	Fieldbus timeout interval, write value	FC3, FC16	Parameter P819: 16-bit value, timeout interval in ms

9.2 *Protocol structure*

The Modbus protocol consists of a header and function code data. The header is the same for all request and response telegrams and error messages (exceptions). A varying number of data is added to it depending on the function code (see following figure).



64064AXX

9.2.1 Header

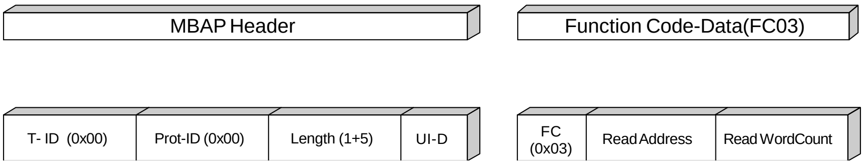
The protocol bytes of the header are described in the following table:

Byte	Designation	Meaning
0	Transaction identifier	Often 0, is simply copied by the server (slave)
1		
2	Protocol identifier	0
3		
4	Length field (upper byte)	0
5	Length field (lower byte)	Number of function codes data bytes + 1 (unit identifier)
6	Unit identifier (slave address)	This is the slave address. In order to access the UFR41B process data, it must be set to "0" (0x00) or 255 (0xFF). The following address assignments apply to the parameter channel access (Offset 200 - 203$_{hex}$): • 0 or 254 for parameters of the UFR41B • 1 - 253 for parameters of a lower-level unit connected to UFR41B. The assignment of unit identifier to the units on the system buses is determined via the routing table on the UFR41B memory card (see section "Appendix").
7	Function code	Requested service
8 ...	Data	Data depending on requested service

- The slave simply copies the transaction identifier (byte 0 and 1). It can help the master to identify related actions.
- The protocol identifier (byte 2 and 3) must always be "0".
- The length bytes (byte 4 and 5) specify the number of bytes occurring in the length field. As the maximum telegram length is 255 bytes, the "upper byte" must be "0".
- The unit identifier (byte 6) can be used for distinguishing between several connected stations (e.g. bridges or gateways). It works as a subaddress that is only used for the parameter access for SEW devices. The process data are always mapped to the unit that is addressed via the unit identifier 0 or FF$_{hex}$.
- The 7 bytes of the header are followed by the function code and the data.

9.2.2 Service FC3 - Read Holding Registers

With the service *FC3 Read holding registers*, you can read a variable number of registers (see following figure).



64065AXX

Example

Request:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Requested service: 3 (Read Holding Register)
8	Reference number (high)	Offset
9	Reference number (low)	Offset
10	Word count (high)	Number of words (register)
11	Word count (low)	Number of words (register)

Response:

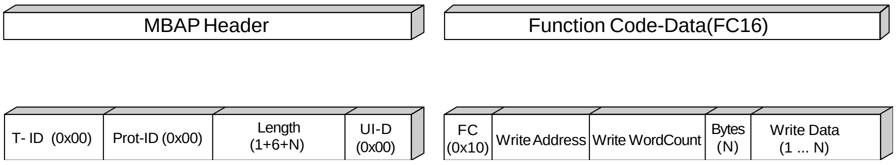
Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Service: 3 (Read Holding Register)
8	Byte count	Number of following bytes
9....	Data	2 - ... Data bytes depending on the length

Exception:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	83_{hex}
8	Exception Code	Error code

9.2.3 Service FC16 - Write Multiple Registers

With the service *FC16 Write Multiple Registers* you can write a variable number of registers (see following figure).



64066AXX

Example

Request:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Requested service: 16 (Write Multiple Registers)
8	Reference number (high)	Offset
9	Reference number (low)	Offset
10	Word count (high)	Number of words (register)
11	Word count (low)	Number of words (register)
12	Byte count	2* Word count
13 ...	Register values	2 - ... Data bytes depending on the length

Response:

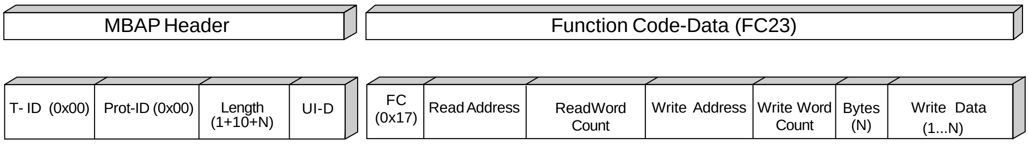
Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Service: 16 (Write Multiple Registers)
8	Reference number (high)	Offset
9	Reference number (low)	Offset
10	Word count (high)	Number of words (register)
11	Word count (low)	Number of words (register)

Exception:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	90_{hex}
8	Exception Code	Error code

9.2.4 Service FC23 - Read/Write Multiple Registers

With the service *FC23 Read/Write Multiple Registers* you can simultaneously write and read a variable number of registers. The write access is carried out first. Preferably, this service is used for the process data (see following picture).



64071AXX

Example

Request:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Requested service: 23 (Read/Write Multiple Registers)
8	Read reference number (high)	Offset
9	Read reference number (low)	Offset
10	Read word count (high)	Number of words (register) always 0
11	Read word count (low)	Number of words (register)
12	Write reference number (high)	Offset
13	Write reference number (low)	Offset
14	Write word count (high)	Number of words (register) always 0
15	Write word count (low)	Number of words (register)
16	Write Byte Count	2* Word count
17 ...	Write register values	2 - ... Data bytes depending on the length

Response:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Service: 23 (Read/Write Multiple Registers)
8	Byte count	Number of following bytes
9	Data	2 - ... Data bytes depending on the length

Exception:

Byte	Designation	Meaning
0 - 6	MBAP header	See chapter "Header".
7	Function code	97_{hex}
8	Exception Code	Error code

9.2.5 Service FC43 - Read Device Identification

The service *FC43 Read device identifications* is also referred to as MEI ("Modbus Encapsulated Interface Transport"). It can tunnel services and method calls. The service *Read Device Identification* is tunneled with the MEI-Type 0x0E. According to the Modbus specification, there are 3 blocks (*Basic*, *Regular* and *Extended*) that can be read. The DHR41B option supports the *Basic* and *Regular* blocks (conformity level 02). The entire block is always read (streaming). This means that values 01 and 02 are permitted in the *Read device ID code*. The *Object ID* must be zero. The response is not fragmented.

Example

Request:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Requested service: 43 (Read Device Identification)
8	MEI type	0x0E
9	Read device ID code	01 or 02
10	Object ID	0

Response:

Byte	Designation	Meaning/permitted values
0 - 6	MBAP header	See chapter "Header".
7	Function code	Service: 43 (Read Device Identification)
8	MEI type	0x0E
9	Read device ID code	01 or 02
10	Conformity level	02
11	More follows	0
12	Next object ID	0
13	Number of objects	e.g. 3
14	Object ID	
15	Object length	
16	Object value	
17	....	

Exception:

Byte	Designation	Meaning
0 - 6	MBAP header	See chapter "Header".
7	Function code	43_{hex}
8	Exception Code	Error code

Objects

UFR41B

Object ID	Name	Type	M/O	Category	Value (example)
0x00	VendorName	ASCII string	Mandatory	Basic	"SEW-EURODRIVE"
0x01	ProductCode				"SEW GATEWAY UFR41B"
0x02	MajorMinorRevisisons				"823 568 0.10" (example)
0x03	VendorUrl		Optional	Regular	"www.sew.de"
0x04	ProductName				"SEW GATEWAY"
0x05	ModelName				"UFR41B"

9.3 Connection management

Up to 8 simultaneous Modbus connections are possible, max. one of which has write access to the process data area (controlling connection).

Connections that are no longer used must be disconnected by the master. If a ninth connection is required and the slave detects an inactive connection, the slave disconnects the inactive connection as it presumes that the corresponding master is no longer active. If there are 8 active connections, the attempt to establish a ninth connection is cancelled (socket is closed on the server). Connections 1 - 8 operate independently of each other. There is no prioritization. Only one controlling connection is permitted that can change the process data.

If a controlling connection has already been established via EtherNet/IP, you cannot establish another controlling connection via Modbus/TCP. The slave can at least buffer one frame on receipt or transmission.

9.3.1 Sending process output data (requesting controlling connection)

Sending process output data is only permitted if the connection already is a controlling connection or if there is no controlling connection. If the unit accepts the connection, it transfers the process output data to the process data image or transmits the process data to a lower-level station (gateway operation). As long as this connection is activated, no other master can change the process output data (PO data).

9.3.2 Disconnecting connections

A connection is deleted from the internal connection list if

- The "keep alive" time has elapsed and the server no longer receives a response, or
- The socket reports a fault
- The connection to the client has been disconnected

  If it was a controlling connection, another controlling connection can be established. If there are no permitted PO data sent within the timeout interval, a fieldbus timeout is triggered.

The "keep alive" time is set to 10 seconds by default. If there is a controlling connection and if the timeout interval is set to more than 5 seconds, the "keep alive" time is increased to twice the value of the timeout interval.

In the event of a cable break or a socket fault of a controlling connection, the fieldbus timeout in the unit will be displayed once the timeout interval has elapsed. Then a new controlling connection can be established.

9.3.3 Timeout monitoring

The timeout monitoring time can be set in the range of 0 to 650 s in steps of 10 ms.

- 0 s and 650 s means: Timeout monitoring is deactivated
- 10 ms - 649.09 s means: Timeout monitoring is activated

The timeout interval can be set via:

- Register object $219E_{hex}$ (8606_{dec})
- Parameter access to index 8606 via register object 200_{hex} 203_{hex}
- Parameters in the parameter tree of MOVITOOLS® MotionStudio

The timeout monitoring is triggered when a controlling connection is activated. The fieldbus driver cyclically checks whether the last PO data update was received within the timeout interval.

If the timeout monitoring is deactivated by setting the timeout interval to 0 or 65000, no fieldbus timeout is detected. This also applies when the controlling connection is dropped.

In the event of a timeout, the timeout response is executed as programmed in the gateway program.

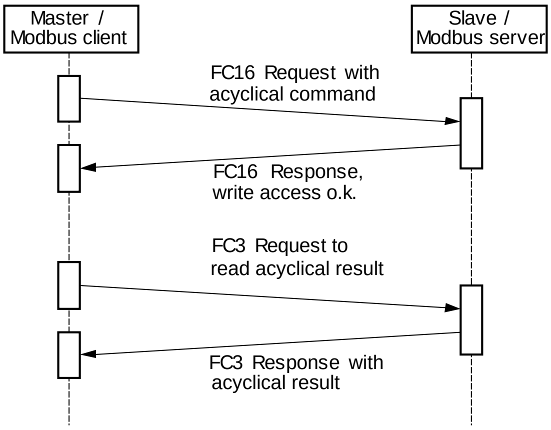
A change to the timeout interval (writing on index 8606) is activated after a re-boot.

9.4 *Parameter access via Modbus/TCP*

A parameter access via the MOVILINK® parameter channel in registers 200_{hex} 203_{hex} via Modbus/TCP requires the services FC3, FC16 or FC23 (write and read access). Write access is used for storing acyclic requests in the corresponding registers. Read services read the responses from the same registers.

This method corresponds to the alternative concept according to the Modbus specifications (appendix A) 'Network Messaging Specification for the MODBUS/TCP Protocol: Version 1.1".

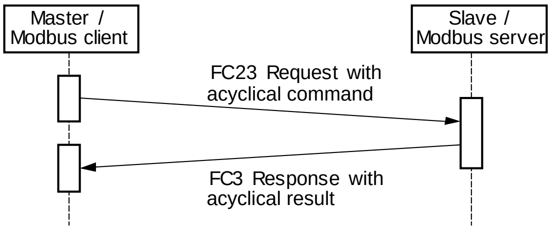
9.4.1 Procedure with FC16 and FC3


Master / Modbus client
Slave / Modbus server
FC16 Request with acyclical command
FC16 Response, write access o.k.
FC3 Request to read acyclical result
FC3 Response with acyclical result

64072AEN

The respective error code is generated if a write access is incorrect (see section "Error codes (exception codes)"). In this way the write services are already processed by sending a *Write Request* (FC16), and the service confirmation can be carried out by evaluating the *Write Response*. Later on, the master will send a *Read Request* (FC03) in order to read out the values that have now been written into the register.

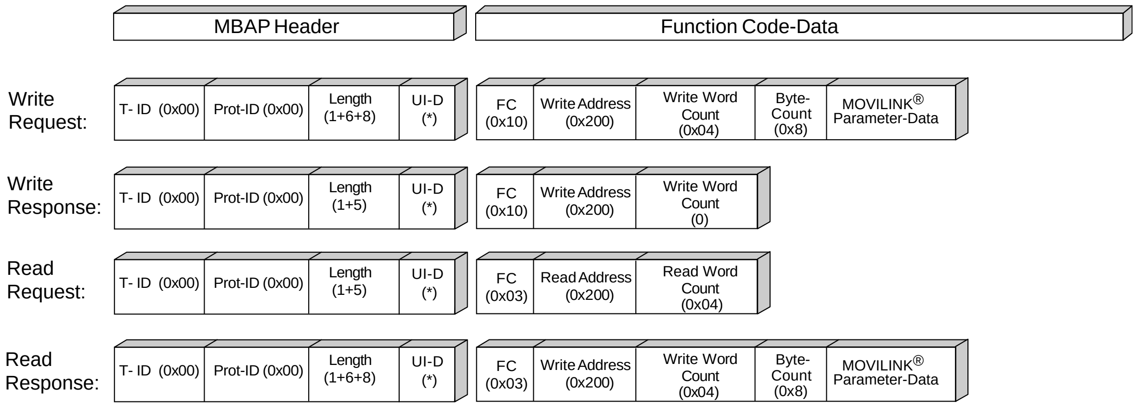
9.4.2 Procedure with FC23


Master / Modbus client
Slave / Modbus server
FC23 Request with acyclical command
FC3 Response with acyclical result

64073AEN

With FC23 the result is returned directly in the response.

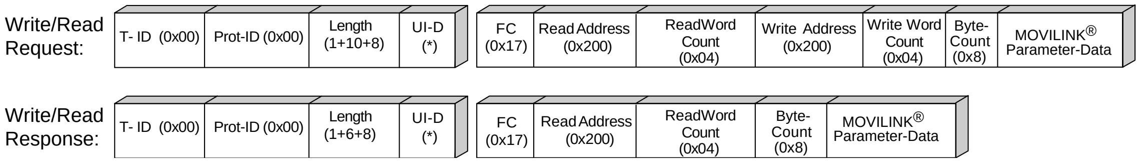
9.4.3 Protocol structure



	MBAP Header				Function Code-Data				
Write Request:	T- ID (0x00)	Prot-ID (0x00)	Length (1+6+8)	UI-D (*)	FC (0x10)	Write Address (0x200)	Write Word Count (0x04)	Byte-Count (0x8)	MOVILINK® Parameter-Data
Write Response:	T- ID (0x00)	Prot-ID (0x00)	Length (1+5)	UI-D (*)	FC (0x10)	Write Address (0x200)	Write Word Count (0)		
Read Request:	T- ID (0x00)	Prot-ID (0x00)	Length (1+5)	UI-D (*)	FC (0x03)	Read Address (0x200)	Read Word Count (0x04)		
Read Response:	T- ID (0x00)	Prot-ID (0x00)	Length (1+6+8)	UI-D (*)	FC (0x03)	Write Address (0x200)	Write Word Count (0x04)	Byte-Count (0x8)	MOVILINK® Parameter-Data

64067AXX

Or:



Write/Read Request:	T- ID (0x00)	Prot-ID (0x00)	Length (1+10+8)	UI-D (*)	FC (0x17)	Read Address (0x200)	ReadWord Count (0x04)	Write Address (0x200)	Write Word Count (0x04)	Byte-Count (0x8)	MOVILINK® Parameter-Data
Write/Read Response:	T- ID (0x00)	Prot-ID (0x00)	Length (1+6+8)	UI-D (*)	FC (0x17)	Read Address (0x200)	ReadWord Count (0x04)	Byte-Count (0x8)	MOVILINK® Parameter-Data		

64165AXX

* The unit identifier (UI-D) is used in gateway operation to map registers 200_{hex} 203_{hex} to the lower-level stations, see section "Header".

The description of the MOVILINK® parameter data (8 bytes) and their mapping to registers 200_{hex} 203_{hex} is described in section "MOVILINK® parameter channel".

9.4.4 MOVILINK® parameter channel

The following table shows the structure of the MOVILINK® acyclic parameter channel. It contains 8 bytes.

Offset	200_{hex}	200_{hex}	201_{hex}	201_{hex}	202_{hex}	202_{hex}	203_{hex}	203_{hex}
Meaning	Management	Subindex	Index high	Index low	MSB data	Data	Data	LSB data
Comment	Management	Parameter index + subindex			4-byte data			
Example: Writing fieldbus timeout (index 8606)	32_{hex}	00_{hex}	21_{hex}	$9E_{hex}$	00_{hex}	00_{hex}	01_{hex}	$F4_{hex}$

You can access the parameter channel with FC3, FC16 and FC23. You can inform the parameter channel of the task in the administration byte using a write access. The task itself is a MOVILINK® service such as *Write*, *Write Volatile* or *Read*. The result can be read with a read access. Refer to the "MOVIDRIVE® MDX60B/61B Communication and Fieldbus Unit Profile" documentation for information on the structure of the parameter channel.®

In this example, the MOVILINK® parameter channel is used to request 500 ms to be written to the fieldbus timeout interval:

- Offset 200 = 3200_{hex} (administration = write 4 bytes / subindex = 0)
- Offset 201 = $219E_{hex}$ (Index = 8606)
- Offset 202 = 0(Data high)
- Offset 203 = $01F4_{hex}$ (data low = 500)

9.5 Error codes (exception codes)

If an error occurs when processing a function code, the Modbus client is informed in an *exception response*. The following *exception codes* can be reported by SEW devices:

Exception Code (hex)	Name	Meaning
01	ILLEGAL FUNCTION	The function code transferred to the request is not supported by the slave unit.
02	ILLEGAL DATA ADDRESS	You have entered an invalid data address for the access to the Modbus slave. This can be due to the following reasons: • Invalid start address when accessing the register of the Modbus slave (not available or the function code cannot be used on this address) • Invalid start address/length combination • No symmetric access with read/write • Wrong object ID (on access via FC43)
03	ILLEGAL DATA VALUE	A part of the data field of the Modbus request contains a value invalid for the Modbus slave. This can be due to the following reasons: • The "Word count" contains an invalid value (smaller than 1 or larger than 125) • The received PDU length is too short or too long (depending on the specified "word count") • Internal error while reading/writing process data
04	SLAVE DEVICE FAILURE	Error while accessing MOVILINK® parameters (e.g. internal timeout)
06	SLAVE DEVICE BUSY	A controlling connection already exists (either via another Modbus controller or another fieldbus system)
0A	GATEWAY PATH UNAVAILABLE	The data cannot be forwarded to a subsystem.

10 Configuring PROFINET IO

10.1 Configuring PROFINET IO controller

The following sections describe the configuration of the UFR41B fieldbus gateway with PROFINET interface. The project planning will be explained using the example of the SIMATIC STEP 7 project planning software and a SIMATIC CPU 315F-2 PN/DP.

10.1.1 Installing the GSD file

TIP

The latest GSD(ML) file version is also available for download on the SEW website.

Proceed as follows to install the GSD file:

1. Start STEP 7 HW Config and select the [Install new GSD file] menu item in the [Extras] menu.

   A window is displayed.
2. Click on [Browse] and select the following file:

   "GSDML-V2.1-DHR41B-UFR41B-JJJJMMTT.xml" (JJJJMMTT represents the date)
3. Click on [OK] to confirm your selection.
4. You will find the PROFINET IO interface for the UFR41 fieldbus gateway via [PROFINET IO]/[Other field units]/[Drives]/[SEW]/[DHR41B/UFR41B] in the hardware catalog.

   2 files are available for configuring the UFR41B option:

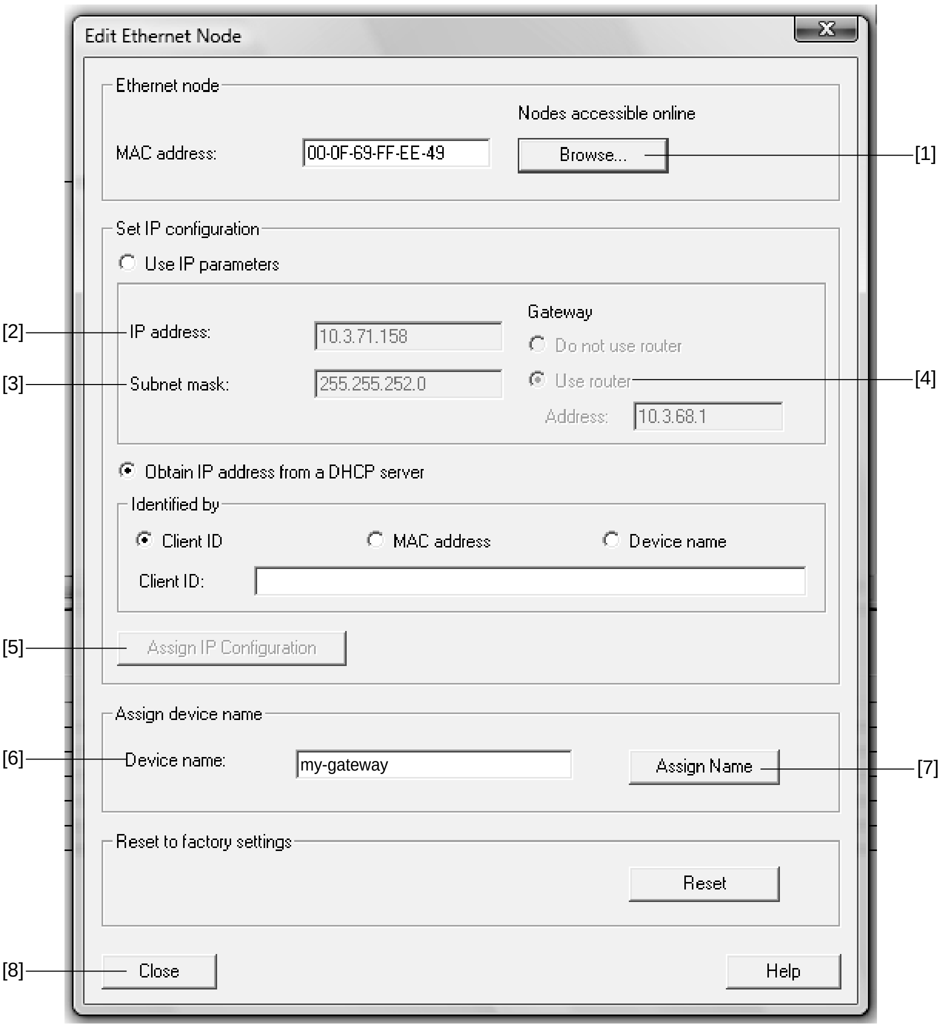
   - UFR41B V1.0
     for controllers that support the PROFINET IO topology detection
   - UFR41B V1.0 **OLD**
     for controllers that do **not** support the PROFINET IO topology detection

10.1.2 Assigning a PROFINET device name

Proceed as follows to assign the PROFINET device name:

1. Select [ETHERNET]/[Edit ETHERNET station] from the [Target system] menu in STEP 7 HW Config.

   The following window opens:



Edit Ethernet Node

Ethernet node
Nodes accessible online
MAC address: 00-0F-69-FF-EE-49 [Browse...]

Set IP configuration
○ Use IP parameters
IP address: 10.3.71.158
Subnet mask: 255.255.252.0
Gateway
○ Do not use router
● Use router
Address: 10.3.68.1
● Obtain IP address from a DHCP server
Identified by
● Client ID ○ MAC address ○ Device name
Client ID:
[Assign IP Configuration]

Assign device name
Device name: my-gateway [Assign Name]

Reset to factory settings
[Reset]

[Close] [Help]

65206AXX

[1] [Browse] button
[2] "IP address" input field
[3] "Subnet mask" input field
[4] "Router address" input field
[5] "Assign IP Configuration" button
[6] "Device name" input field
[7] "Assign name" button
[8] [Close] button

2. Click on the [Browse] [1] button in the "ETHERNET stations" group. You receive an overview of all PROFINET IO stations that you can reach online with your configuration tool.

3. Choose the required station.

   The SEW station appears under unit type. The default device name is "PNETDeviceName" and must be changed appropriately. Several UFR41B gateways can be distinguished by the MAC addresses displayed. The MAC address is affixed to the UFR41B fieldbus gateway.

4. Enter the device name in the "Device name" input field [6] and click the [Assign name] button [7].

   The device name can have up to 255 characters. The device name is transferred to and saved in the station.

   You can reset the device name of the UFR41B fieldbus gateway online using the [Reset] button. This requires a restart of the UFR41B option.

5. Specify an IP address [2] and a subnet mask [3] as well as a router address [4] if required.

   Click the [Assign IP configuration] button [5].

TIP

The IO controller must not yet be in a cyclic data transmission with the IO devices.

6. Check whether the settings have been applied by once again clicking the [Browse] button [1].

7. Click on the [Close] button [8].

10.2 Configuring the PROFINET connection for the UFR41B fieldbus gateway

10.2.1 Creating a new project

Proceed as follows to create a new project:

1. Start the SIMATIC Manager and create a new project.

   Select your control type and add the required modules. The following modules make sense:

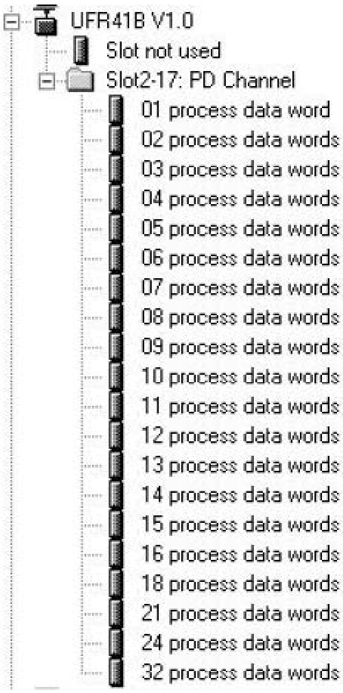
   - **OB82 module:**This module makes sure that the controller does not trigger a "STOP" in the event of so-called diagnostic alarms.
   - **OB86 module:** This module indicates the failure of decentralized peripherals.
   - **OB122 module:**This module is addressed if the controller cannot access data of a station of the decentralized periphery. This can occur, for example, when MOVI-PLC® *advanced* DHR41B is ready for operation later than the control system.
2. Start STEP 7 HW Config and select the PROFINET IO slot in the control rack.
3. Add a PROFINET IO system by right-clicking the context menu with your mouse.
4. Specify an IP address for the PROFINET IO controller when doing this.
5. Add a new PROFINET subsystem using the [ETHERNET] button.
6. Open [PROFINET IO]/[Additional field devices]/[Drives]/[SEW]/[DHR41B/UFR41B] in the hardware catalog.

   2 entries are available for the DHR41B option:

   - UFR41B V1.0
     for controllers that support the PROFINET IO topology detection
   - UFR41B V1.0 **OLD**
     for controllers that do **not** support the PROFINET IO topology detection

7. Move the entry "DHR41B/UFR41B" to the PROFINET IO system using the mouse and assign a PROFINET station name. This name must correspond to the PROFINET device name specified in the UFR41B fieldbus gateway.
8. Enter the IO and periphery addresses in slot 2 and save the configuration.

   The slot model is used for configuration with PROFINET. Each slot is assigned to a gateway fieldbus interface. The following structure is used:



- UFR41B V1.0
  - Slot not used
  - Slot2-17: PD Channel
    - 01 process data word
    - 02 process data words
    - 03 process data words
    - 04 process data words
    - 05 process data words
    - 06 process data words
    - 07 process data words
    - 08 process data words
    - 09 process data words
    - 10 process data words
    - 11 process data words
    - 12 process data words
    - 13 process data words
    - 14 process data words
    - 15 process data words
    - 16 process data words
    - 18 process data words
    - 21 process data words
    - 24 process data words
    - 32 process data words

12125AXX

   The default configuration "Slot not used" must not be modified. Slot 1 is reserved for future PROFIsafe applications.

   Slots 2 to 17 can be assigned process data channels. The maximum process data length is 2 x 32 words.

9. Extend your user program by data exchange with the new units. Process data transfer is consistent. SFC14 and SFC15 can be used to transfer process data.

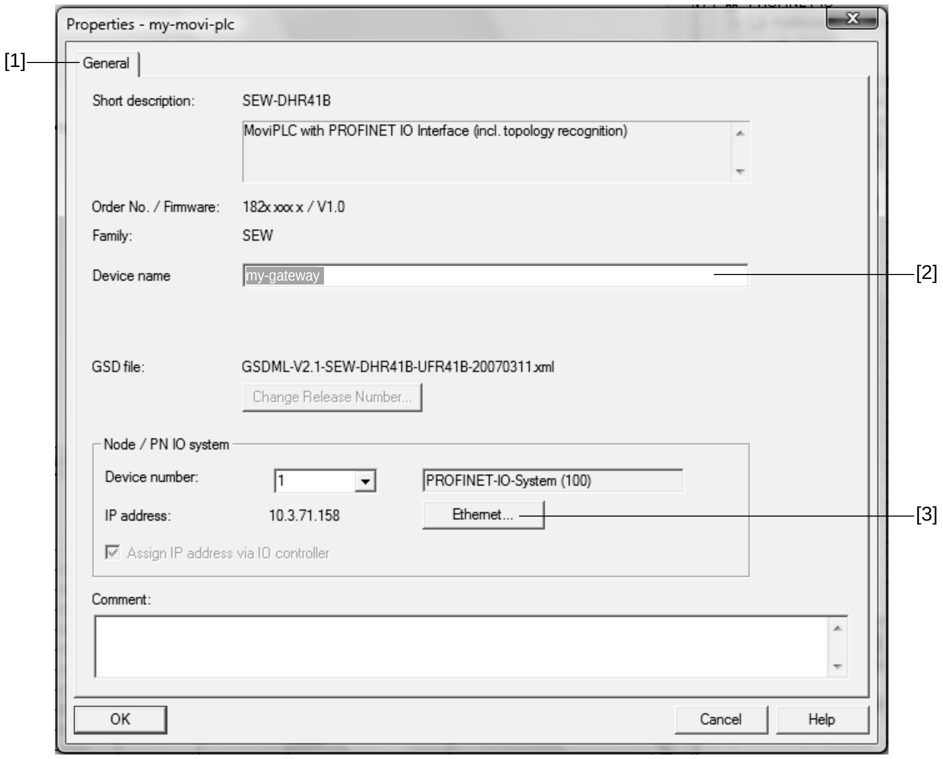
10.2.2 Configuring a station

When the individual slots are configured, the new station has to be configured with further settings.

Proceed as follows to configure a station:

1. Double-click on the unit symbol of the new station.

   The following window opens:



Properties - my-movi-plc

General

Short description: SEW-DHR41B

MoviPLC with PROFINET IO Interface (incl. topology recognition)

Order No. / Firmware: 182x xxx x / V1.0

Family: SEW

Device name: my-gateway

GSD file: GSDML-V2.1-SEW-DHR41B-UFR41B-20070311.xml

Change Release Number...

Node / PN IO system

Device number: 1 PROFINET-IO-System (100)

IP address: 10.3.71.158 Ethernet...

☑ Assign IP address via IO controller

Comment:

OK Cancel Help

65202AXX

[1] "General" tab

[2] "Device name" input field

[3] [ETHERNET] button

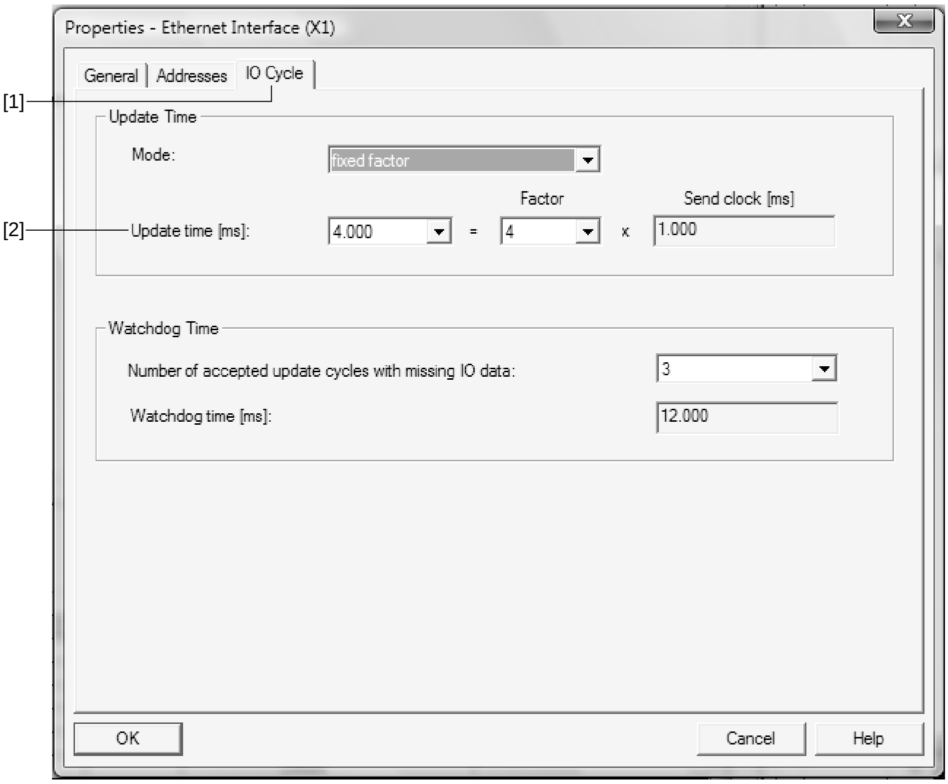
2. Enter the device name assigned before in the "Device name" input field [2] on the "General" tab [1].

   Note that the name is case-sensitive.

3. Click the [ETHERNET] button [3] in the "Station/PN IO system" group to enter the previously assigned IP address.

4. Double-click on the "ETHERNET interface" slot in order to set the station’s update time.

   The following window opens:



Update Time

Mode: fixed factor

Update time [ms]: 4.000 = Factor 4 x Send clock [ms] 1.000

Watchdog Time

Number of accepted update cycles with missing IO data: 3

Watchdog time [ms]: 12.000

64632AXX

[1] "IO Cycle" tab page

[2] "Update time" selection field

5. On the "IO cycle" tab [1], set the update time [2] for the station to update its process data.

   The UFR41B fieldbus gateway support a minimum update time of 4 ms.

10.3 PROFINET configuration with topology detection

10.3.1 Introduction

The PROFINET technology detection allows for projecting and monitoring the structure of the network with the PROFINET IO controller in addition to the PROFINET IO devices.

The so-called "Physical device (PHDEV)" is the starting point for the configuration. The PDEV is a model for the ETHERNET interface and can be found in slot 0 of the project planning with an "ETHERNET interface" subslot and one subslot for each ETHERNET port.

The visible ETHERNET ports can be connected to the configuration tool. The result is an image of the desired ETHERNET routing for the plant. This image is stored in the PROFINET IO controller.

In order to be able to determine the real plant topology, the PROFINET IO devices must support the so-called LLDP protocol. The PROFINET IO devices exchange information with the neighboring PROFINET IO devices via LLDP. Via LLDP, each PROFINET IO device cyclically sends information on its own PROFINET device name and port number. The neighboring unit receives and stores this information. Now a PROFINET IO controller can read the stored information from the PROFINET IO devices, in this way determining the real plant topology.

By comparing the projected topology with the real topology, you can detect any missing or incorrectly wired PROFINET IO devices and locate them in the plant.

Apart from cabling you can still determine the transmission characteristics for the ports. For example, you can set an "Auto negotiation" port to "100 Mbit full duplex". The settings will be monitored.

SNMP as a protocol for network diagnostics extends the topology detection with standard diagnostics mechanisms from the IT area.

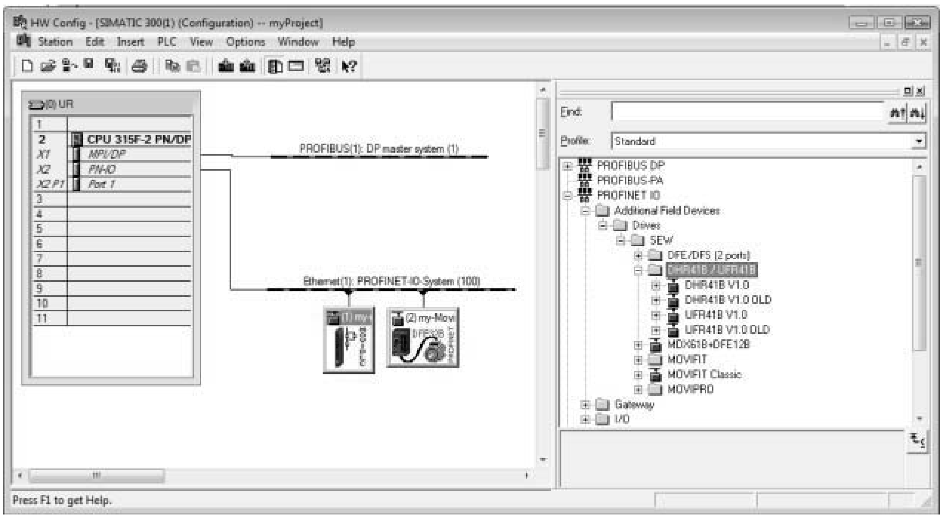
10.3.2 Configuring the PROFINET topology

The configuration procedure for a PROFINET topology will be described using the example of SIMATIC STEP 7. There are various approaches for configuration in SIMATIC STEP 7. This example will focus on one approach.

1. In STEP 7 HW Config, import the PROFINET devices from the hardware catalog into the PROFINET network as usual.

   Make sure that the PROFINET IO controller supports the topology detection. The controller manufacturer will provide relevant information.

   The hardware catalog contains several entries for each SEW interface marked as different versions. An entry marked with "OLD" does not support PROFINET IO topology detection.

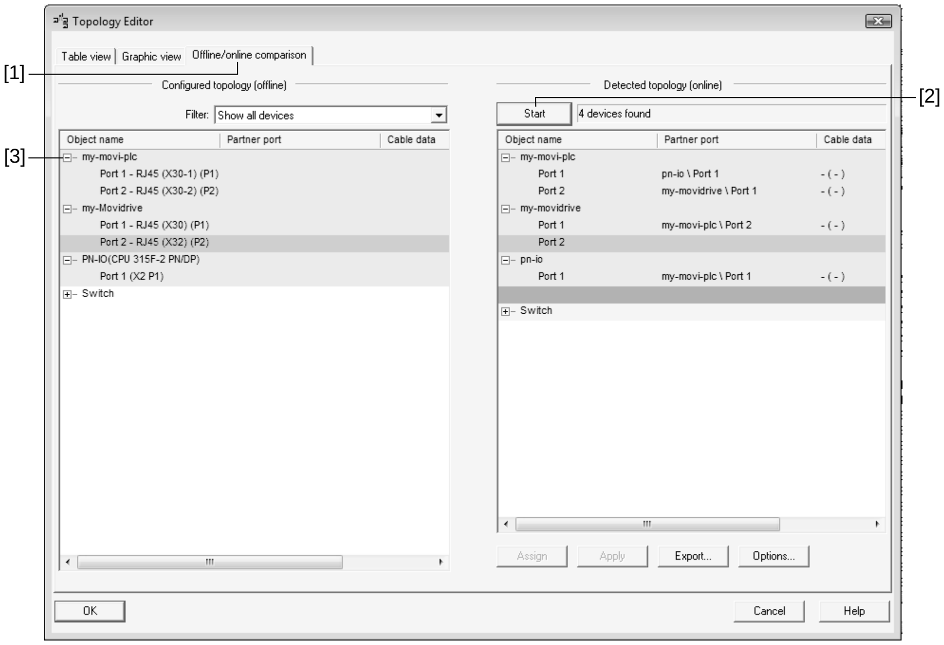


12099AXX

2. Right-click on the "PROFINET IO system" and select "PROFINET IO topology" from the context menu.

   The "Topology editor" window is displayed.

3. Select the "Offline/online comparison" tab [1].



Configured topology (offline)

Filter: Show all devices

Object name	Partner port	Cable data
⊟ my-movi-plc		
Port 1 - RJ45 (X30-1) (P1)		
Port 2 - RJ45 (X30-2) (P2)		
⊟ my-Movidrive		
Port 1 - RJ45 (X30) (P1)		
Port 2 - RJ45 (X32) (P2)		
⊟ PN-IO(CPU 315F-2 PN/DP)		
Port 1 (X2 P1)		
⊞ Switch		

Detected topology (online)

4 devices found

Object name	Partner port	Cable data
⊟ my-movi-plc		
Port 1	pn-io \ Port 1	- (-)
Port 2	my-movidrive \ Port 1	- (-)
⊟ my-movidrive		
Port 1	my-movi-plc \ Port 2	- (-)
Port 2		
⊟ pn-io		
Port 1	my-movi-plc \ Port 1	- (-)
⊞ Switch		

64633AXX

[1] "Offline/online comparison" tab

[2] [Start] button

[3] Plus/minus symbol

4. Determine the online topology by clicking [Start] [2].
5. Make sure that the determined topology complies with your requirements by clicking on the plus symbol [3] in the online topology and checking the partner port.

   The following units are displayed in this example:

   - 2 SEW units (MOVIDRIVE®, MOVI-PLC® *advanced* DHR41B)
   - One controller
   - One switch

   The switch does not support topology and is highlighted white. The remaining PROFINET IO devices are not linked yet and are therefore highlighted yellow.
6. In order to apply the determined online topology to the configuration port by port, right-click on a port. From the context menu, select "Apply port interconnection". Repeat this procedure for all ports of the devices until the lists are green.

10.3.3 Changing the port properties

The two ETHERNET ports of the PROFINET interface are set to "Automatic setup" by default. Observe the following for this default setup:

- Auto-negotiation and auto-crossover are activated in this setup.
- The baud rate and the duplex mode are configured automatically.
- The neighboring port must also be set to "Automatic setup".
- You can use patch or crossover cables.

You may set a port to "100 Mbit/s full duplex". Observe the following for this setup:

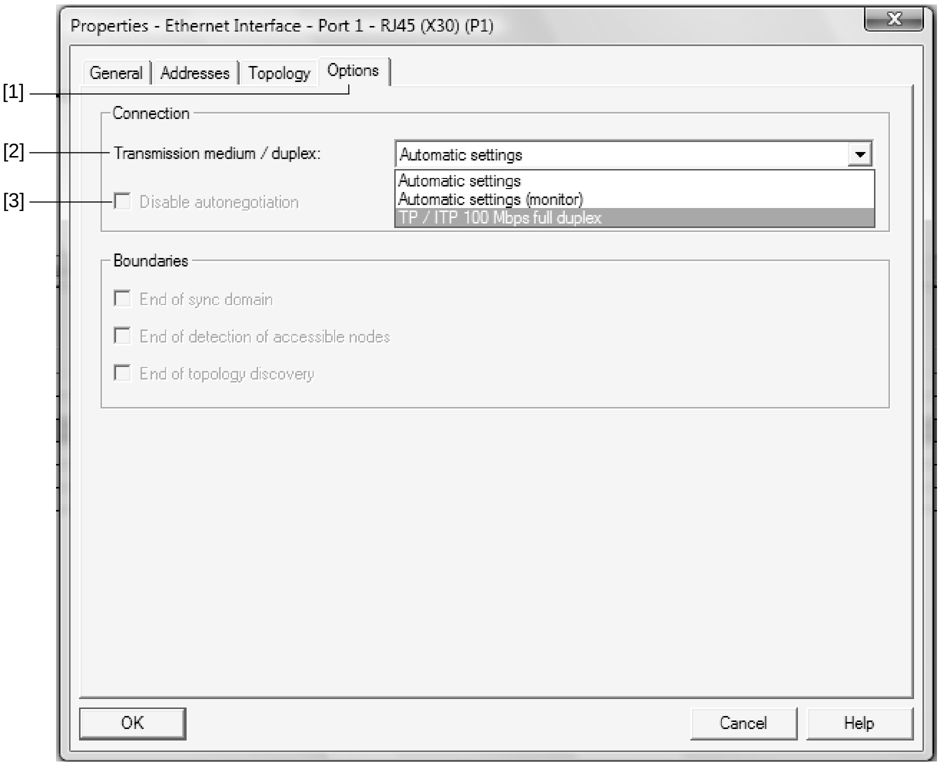
- This setting must also be made for the port of the neighbor unit, otherwise it would work with 100 Mbit/s half duplex.
- If auto-crossover is deactivated, you have to use cross cables.

Proceed as follows to set a port to "100 Mbit/s full duplex":

1. Select a unit in STEP 7 HW Config.
2. Select the desired port on slot 0.
3. Right-click on the port and select "Object properties" from the context menu.

   A window is displayed.

4. Select the "Options" tab [1].



Properties - Ethernet Interface - Port 1 - RJ45 (X30) (P1)

[1]
[2]
[3]
General
Addresses
Topology
Options
Connection
Transmission medium / duplex:
Automatic settings
Automatic settings
Automatic settings (monitor)
TP / ITP 100 Mbps full duplex
□ Disable autonegotiation
Boundaries
□ End of sync domain
□ End of detection of accessible nodes
□ End of topology discovery
OK
Cancel
Help

64634AXX

[1] "Options" tab

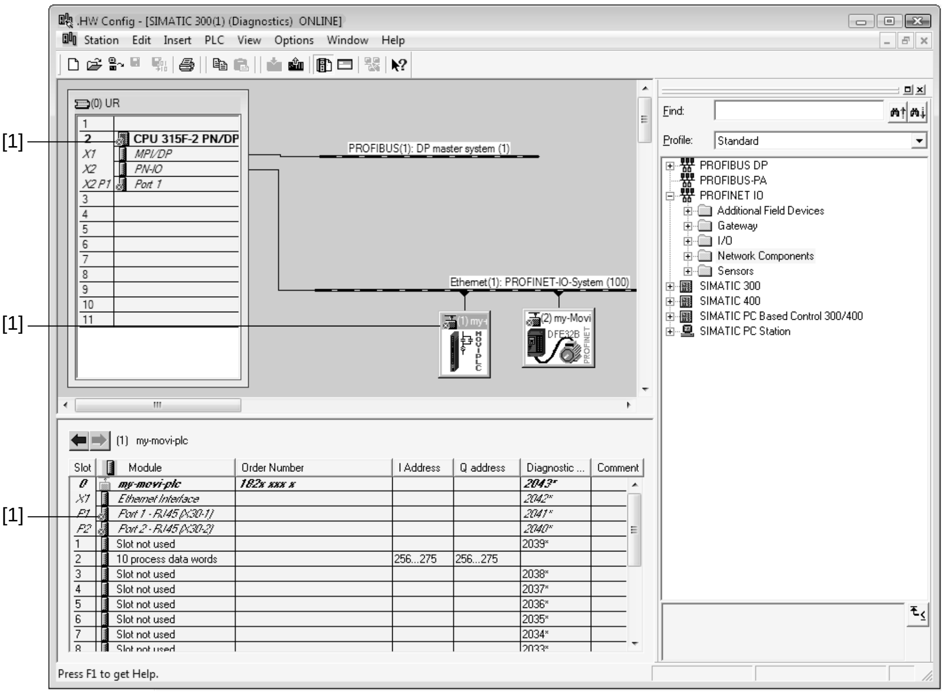
[2] "Transmission medium/duplex" selection list

[3] "Auto-negotiation/auto-crossover" checkbox

5. From the "Transmission medium/duplex" list [2], select "TP/ITP with 100 Mbit/s full duplex".
6. Deactivate the "Auto-negotiation/auto-crossover" checkbox [3].

10.3.4 Topology diagnostics

Topology errors are reported to the PROFINET IO controller as diagnostics alarms. In the event of an error, the EXTF-LED of the PROFINET IO controller is lit. The error is also indicated by a red cross [1] in STEP 7 HW Config.



(1) my-movi-plc

Slot	Module	Order Number	I Address	Q address	Diagnostic ...	Comment
0	my-movi-plc	182x xxx x			2043*	
X1	Ethernet Interface				2042*	
P1	Port 1 - RJ45 (X30-1)				2041*	
P2	Port 2 - RJ45 (X30-2)				2040*	
1	Slot not used				2039*	
2	10 process data words		256...275	256...275		
3	Slot not used				2038*	
4	Slot not used				2037*	
5	Slot not used				2036*	
6	Slot not used				2035*	
7	Slot not used				2034*	
8	Slot not used				2033*	

Press F1 to get Help.

64635AXX

[1] "Red cross" symbol for error

Possible causes:

- ETHERNET ports mixed up
- Wrong port property settings
- Units cannot be addressed

Proceed as follows to display information on an error:

1. Select the unit or the respective slot.
2. Right-click on it and select "Component status" from the context menu.
   A window is displayed.
3. Select the "Communication diagnostics" tab.

10.3.5 Port statistics

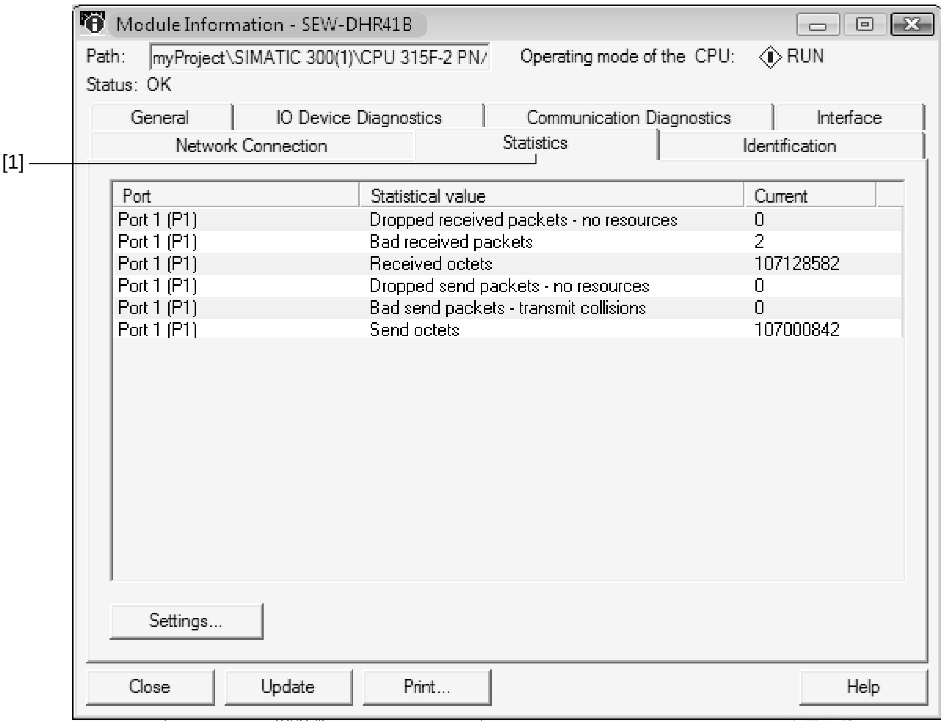
Proceed as follows to display the port statistics for an ETHERNET port in STEP 7 HW Config:

1. Click the "ONLINE ↔ OFFLINE" symbol to switch to "Online" communication mode.
2. Select a unit.
3. Select the desired port on slot 0.
4. Right-click on it and select "Component status" from the context menu.

   A window is displayed.

   Select the "Statistics" tab [1].

   The following view is displayed:



Path: myProject\SIMATIC 300(1)\CPU 315F-2 PN/ Operating mode of the CPU: RUN
Status: OK

Port	Statistical value	Current
Port 1 (P1)	Dropped received packets - no resources	0
Port 1 (P1)	Bad received packets	2
Port 1 (P1)	Received octets	107128582
Port 1 (P1)	Dropped send packets - no resources	0
Port 1 (P1)	Bad send packets - transmit collisions	0
Port 1 (P1)	Send octets	107000842

64345AXX

[1] "Statistics" tab

The following statistic values can be displayed:

- **Dropped received packets – no resources** indicates the number of valid ETHERNET packets dropped on receipt. A large number of dropped valid packets suggests a high load on the bus system. In this case, try to reduce the utilization by especially reducing the number of broadcast and multicast telegrams and reducing the IO cycle or the number of PROFINET units in a line if required.

- **Bad received packets** indicates the number of faulty ETHERNET packets. A high number suggests a bus fault. In this case, check the cabling and shielding of the network.
- **Received octets** indicates the number of received packets.
- **Dropped sent packets – no resources** indicates the number of valid ETHERNET packets dropped when being sent. A large number of dropped valid packets suggests a high load on the bus system. In this case, try to reduce the utilization by especially reducing the number of broadcast and multicast telegrams and reducing the IO cycle or the number of PROFINET units in a line if required.
- **Bad sent packets transmission collisions** indicates the number of ETHERNET packets dropped due to collisions. There should be no collisions in a switched network.
- **Sent Octets** indicates the number of sent packets.

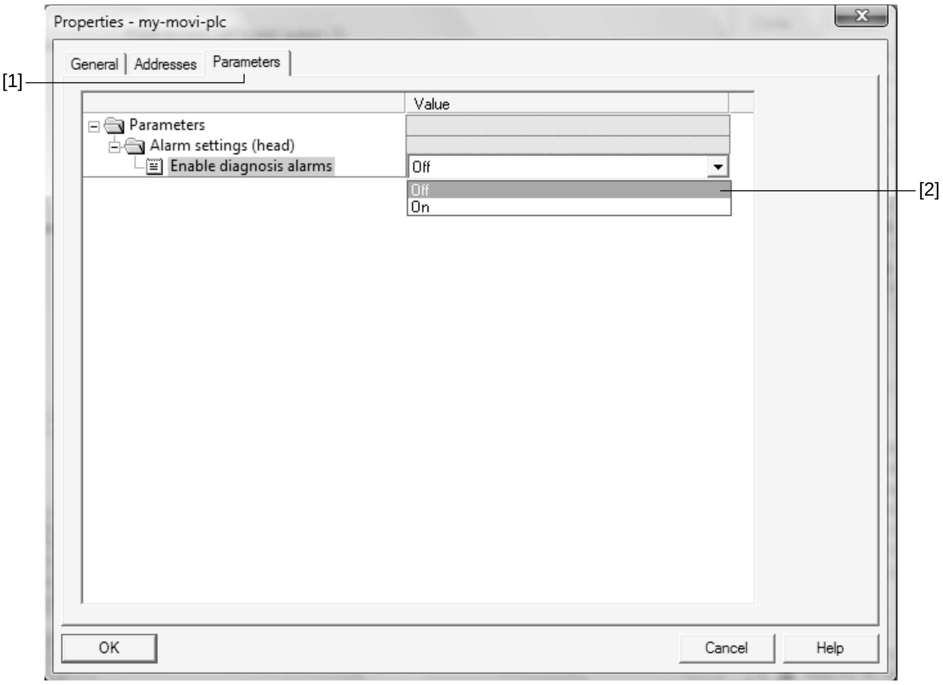
10.4 PROFINET diagnostic alarms

10.4.1 Activating diagnostic alarms

The PROFINET interface supports diagnostic alarms in the event of a unit fault. These diagnostic alarms are deactivated by default. Proceed as follows to activate the diagnostic alarms in STEP 7 HW Config:

1. Highlight slot 0.
2. Right-click on it and select "Object properties ..." from the context menu.

   A window is displayed.
3. Select the "Parameters" tab [1].
4. In "Activate diagnostics alarms" [2], set the alarms to "ON".



64646AXX

[1] "Parameters" tab

[2] "Activate diagnostic alarms" node

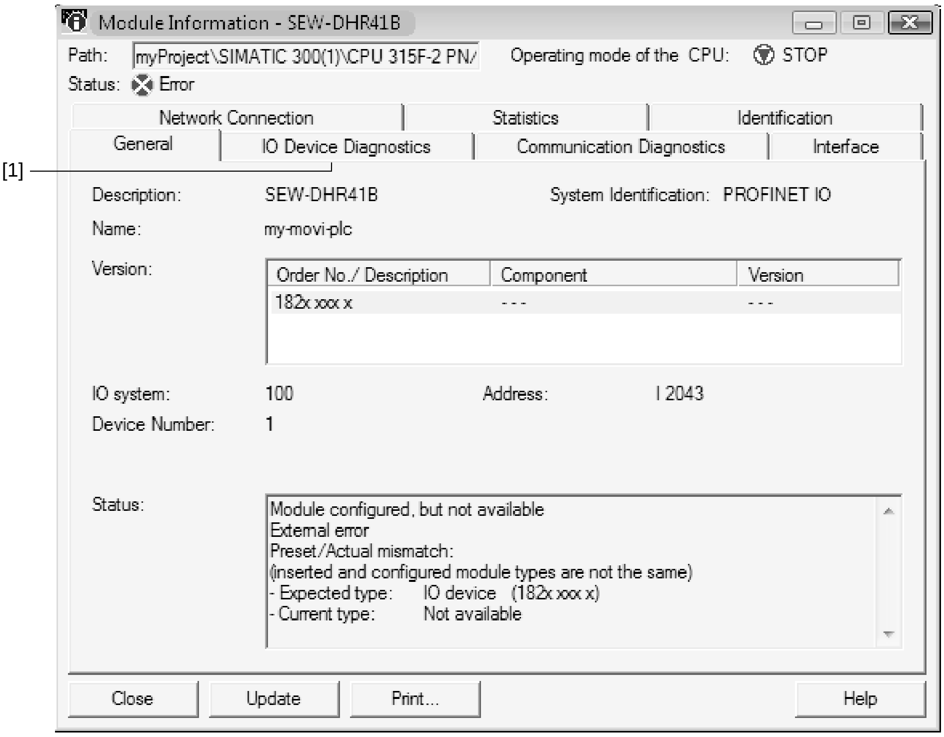
10.4.2 Determining the cause of an error

An error in the function unit belonging to the plug-in module causes a diagnostic alarm to be sent to the controller as an "incoming event".

Proceed as follows to determine an error in STEP 7 HW Config:

1. Click the "ONLINE ↔ OFFLINE" symbol to switch to the "Online" communication mode.
2. Mark the symbol of the SEW PROFINET interface.
3. Right-click on it and select "Component status" from the context menu.

   A window is displayed.



Path: myProject\SIMATIC 300(1)\CPU 315F-2 PN/ Operating mode of the CPU: STOP
Status: Error

Description: SEW-DHR41B System Identification: PROFINET IO
Name: my-movi-plc
Version:

Order No./ Description	Component	Version
182x xxx x	- - -	- - -

IO system: 100 Address: I 2043
Device Number: 1

Status:
Module configured, but not available
External error
Preset/Actual mismatch:
(inserted and configured module types are not the same)
- Expected type: IO device (182x xxx x)
- Current type: Not available

64647AXX

[1] "IO device diagnostics" tab

4. Select the "IO device diagnostics" tab [1].
5. Click on [Display] to receive detailed information on the fault.

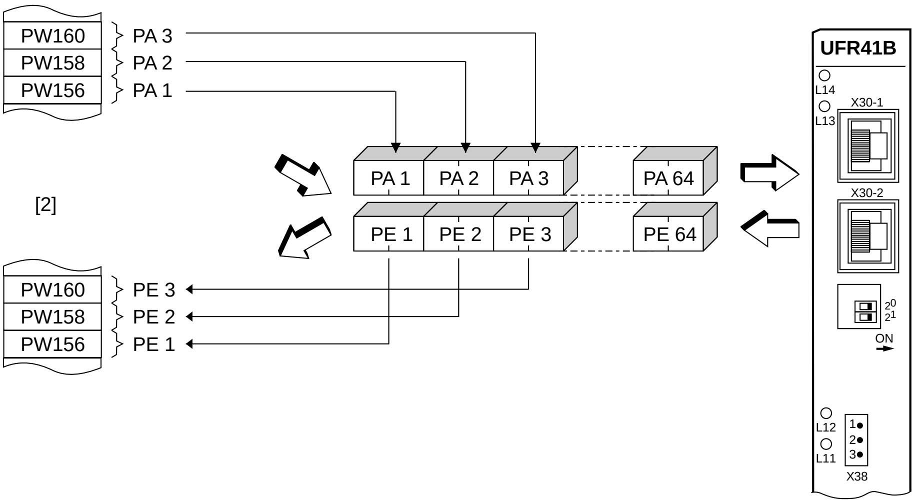
   After resetting the fault, a so-called "ongoing event" is sent to the controller. The SF LED of the CPU goes out and no more faults are displayed in the module information.

11 Operating Characteristics (PROFINET IO)

This chapter describes the basic characteristics of the UFR41B fieldbus gateway on the PROFINET system.

11.1 Process data exchange with the UFR41B fieldbus gateway

The UFR41B fieldbus gateway is controlled via the slave units connected to it using the process data channel, which is up to 2 x 32 I/O words in length.



65201AXX

Figure 2: Mapping PROFINET data in the PLC address range

[1] Address range of the higher-level PLC

PI1 - PI64	Process input data
PO1 - PO64	Process output data

Control example for Simatic S7

Depending on the chosen process data configuration, process data are exchanged with the UFR41B fieldbus gateway via SIMATIC S7 either directly using load and transfer commands, or using the special system functions *SFC 14 DPRD_DAT* and *SFC15 DPWR_DAT*.

STEP7 example program

In this example, the UFR41B fieldbus gateway is configured with the process data configuration *10 PD* to the input addresses PIW512... and output addresses POW512... .

A data block DB3 is created with about 50 data words.

When SFC14 is called, the process input data is copied to data block DB3, data words 0 to 18. When SFC15 is called after the control program has been processed, the process output data are copied from data words 20 - 38 to the output address POW 512 ...

Note the length information in bytes for the *RECORD* parameter. The length information must correspond to the configured length.

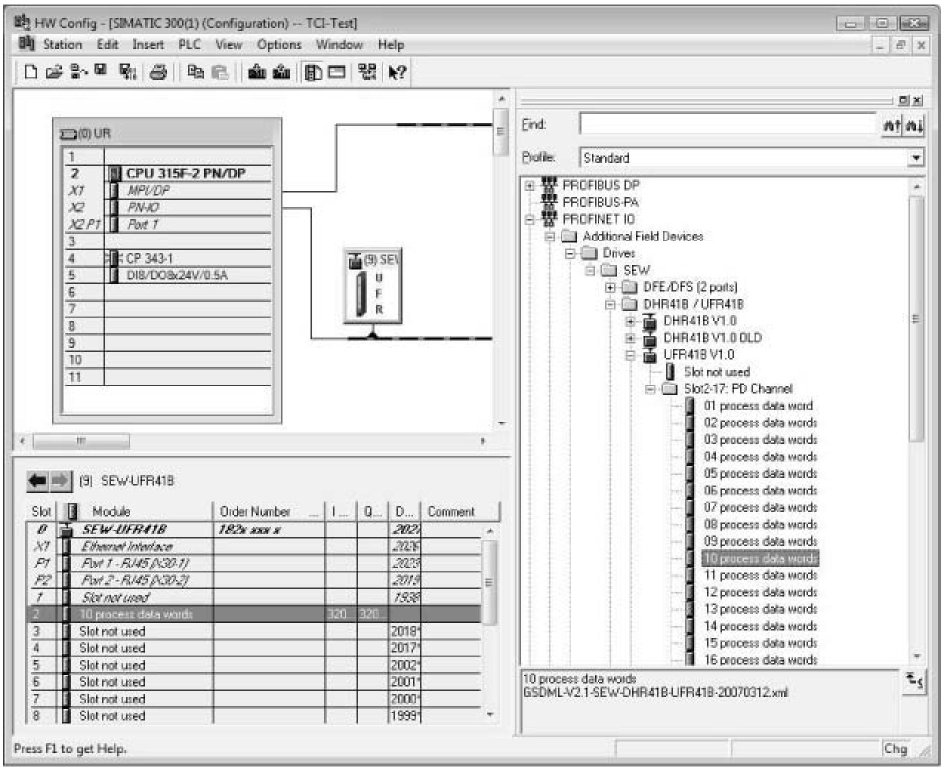
Refer to the online help for STEP7 for further information about the system functions.

```
//Start of cyclical program processing in OB1
BEGIN
NETWORK
TITLE = Copy PI data from the DHR41B control card to DB3, words 0...18
CALL SFC  14 (DPRD_DAT)    //Read DP slave record
  LADDR  := W#16#240 //Input address 512
  RET_VAL:= MW 30          //Result in flag word 30
  RECORD := P#DB3.DBX 0.0 BYTE 20 //Pointer

NETWORK
TITLE =PLC program with drive application
// PLC program uses the process data in DB3 for data exchange
// with the UFR41B control card

L  DB3.DBW 0   //Load PE1
L  DB3.DBW 2   //Load PE2
L  DB3.DBW 4   //Load PE3
// etc.

L  W#16#0006
T DB3.DBW 20   //Write 6hex to PO1
L 1500
T DB3.DBW 22   //Write 1500dec to PO2
L  W#16#0000
T DB3.DBW 24   //Write 0hex to PO3
// etc.

NETWORK
TITLE = Copy PO data from DB3, words 20...38 to the DHR41B control card
CALL SFC  15 (DPWR_DAT)             //Write DP slave record
  LADDR  := W#16#200                //Output address 512 = 200hex
  RECORD := P#DB3.DBX 20.0 BYTE 20  //Pointer to DB/DW
  RET_VAL:= MW 32                   //Result in flag word 32
```

TIP

- This sample program is a free service that demonstrates only the basic approach to generating a PLC program as a non-binding sample. SEW is not liable for the contents of the sample program.
- You can download S7 sample projects from the SEW homepage (http://www.sew-eurodrive.de) under "Software".

The following figure shows the corresponding configuration of the UFR41B fieldbus gateway in the hardware configuration of STEP7.


HW Config - [SIMATIC 300(1) (Configuration) -- TCI-Test]
Station Edit Insert PLC View Options Window Help
(0) UR
(9) SEW
U
F
R
Find:
Profile: Standard
Press F1 to get Help.
Chg

1		
2		CPU 315F-2 PN/DP
X1		MPI/DP
X2		PN-IO
X2 P1		Port 1
3		
4		CP 343-1
5		DI8/DO8x24V/0.5A
6		
7		
8		
9		
10		
11		

- PROFIBUS DP
- PROFIBUS-PA
- PROFINET IO
  - Additional Field Devices
    - Drives
      - SEW
        - DFE/DFS (2 ports)
        - DHR41B / UFR41B
          - DHR41B V1.0
          - DHR41B V1.0 OLD
          - UFR41B V1.0
            - Slot not used
            - Slot2-17: PD Channel
              - 01 process data word
              - 02 process data words
              - 03 process data words
              - 04 process data words
              - 05 process data words
              - 06 process data words
              - 07 process data words
              - 08 process data words
              - 09 process data words
              - 10 process data words
              - 11 process data words
              - 12 process data words
              - 13 process data words
              - 14 process data words
              - 15 process data words
              - 16 process data words

10 process data words
GSDML-V2.1-SEW-DHR41B-UFR41B-20070312.xml

(9) SEW-UFR41B

Slot		Module	Order Number	I ...	Q...	D...	Comment
0		SEW-UFR41B	182x xxx x			202	
X1		Ethernet Interface				202	
P1		Port 1 - RJ45 (X30-1)				202	
P2		Port 2 - RJ45 (X30-2)				201	
1		Slot not used				1938	
2		10 process data words		320...	320...		
3		Slot not used				2018*	
4		Slot not used				2017*	
5		Slot not used				2002*	
6		Slot not used				2001*	
7		Slot not used				2000*	
8		Slot not used				1999*	

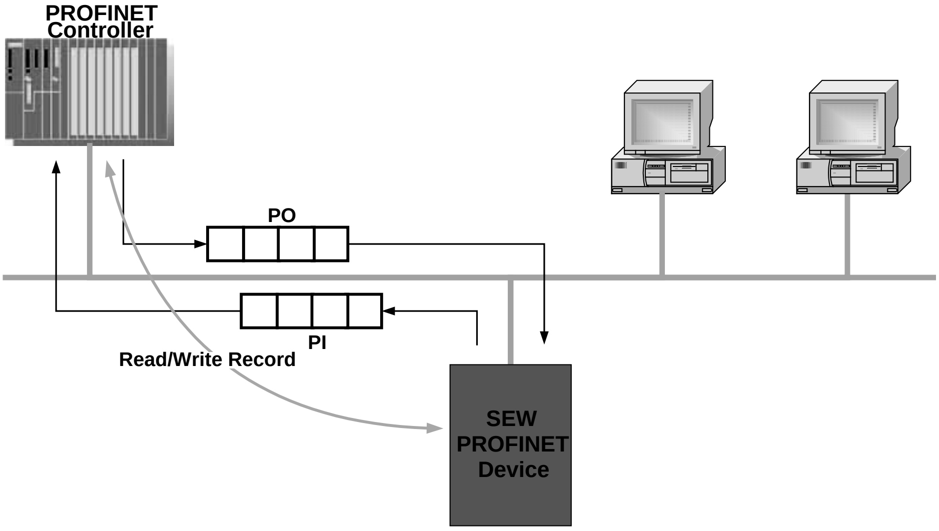
12124AXX

11.2 Parameterization via PROFIdrive dataset 47

	TIP
	The S7 sample project "MOVILINK® parameter channel" can be used for this parameter channel. You can download it from the SEW homepage (http://www.sew-eurodrive.de) under "Software".

11.2.1 PROFINET data records

With "Read Record" and "Write Record", PROFINET offers acyclic services that can be used to transfer parameter data between PROFINET controller (master) and a PROFINET device (slave). Via UDP (User Datagram Protocol), the priority of this data exchange is lower than the priority of the process data exchange.



62204AXX

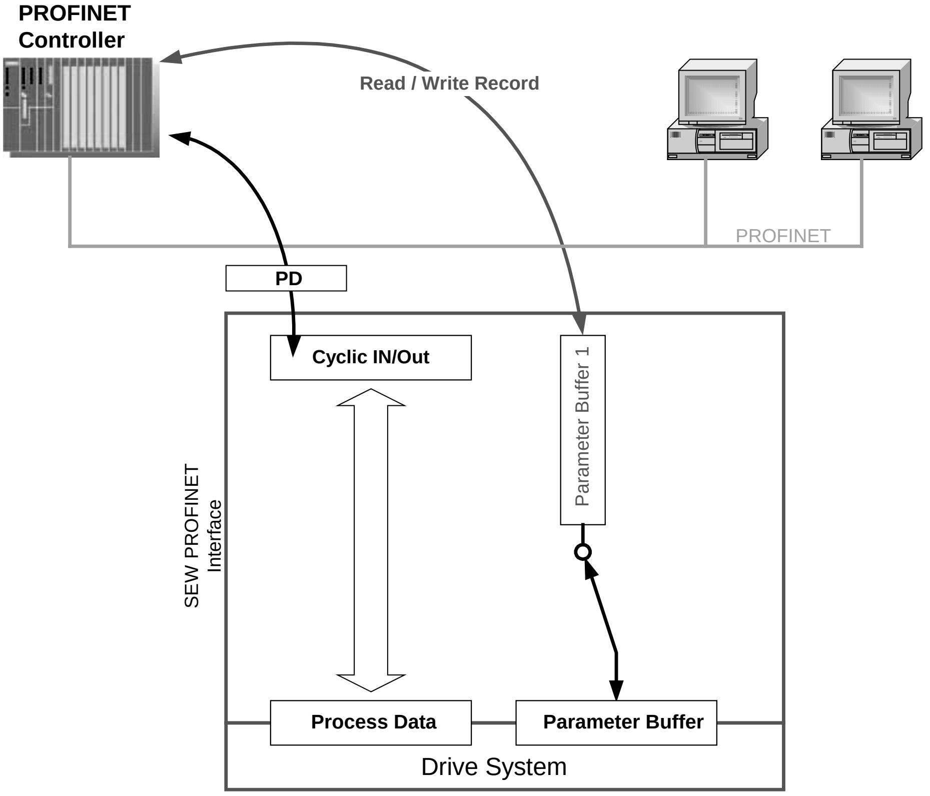
The user data transported via an acyclic PROFINET service is grouped in a data record. Each data record is clearly addressed by the following characteristics:

- API
- Slot number
- Subslot number
- Index

The structure of data record 47 is used for the parameter exchange with SEW-EURODRIVE PROFINET units. The structure of data record 47 is specified in the PROFIdrive profile drive technology of the PROFIBUS user organization as of V4.0 as PROFINET parameter channel. Different procedures for accessing parameter data of the SEW-EURODRIVE PROFINET unit are provided via this parameter channel.

Characteristics of the SEW-EURODRIVE PROFINET devices

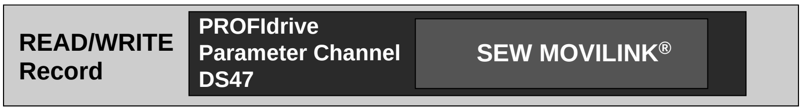
The SEW-EURODRIVE PROFINET units that support acyclic Read Record and Write Record services all have the same communication characteristics. The units are basically controlled via a PROFINET controller with cyclic process data. Additionally, this controller (usually a PLC) can set the parameters for the SEW-EURODRIVE PROFINET unit via Read Record and Write Record.



62205AXX

11.2.2 Structure of the \PROFINET parameter channel

Generally, the parameter setting of the devices to the PROFIdrive-Base Mode Parameter Access of profile version 4.0 is implemented via data record 47. The *Request ID* entry is used to distinguish between parameter access based on PROFIdrive profile or via SEW-MOVILINK® services. The following table shows the possible codes of the individual elements. The data record structure is the same for PROFIdrive and MOVILINK® access.



62206AXX

The following MOVILINK® services are supported:

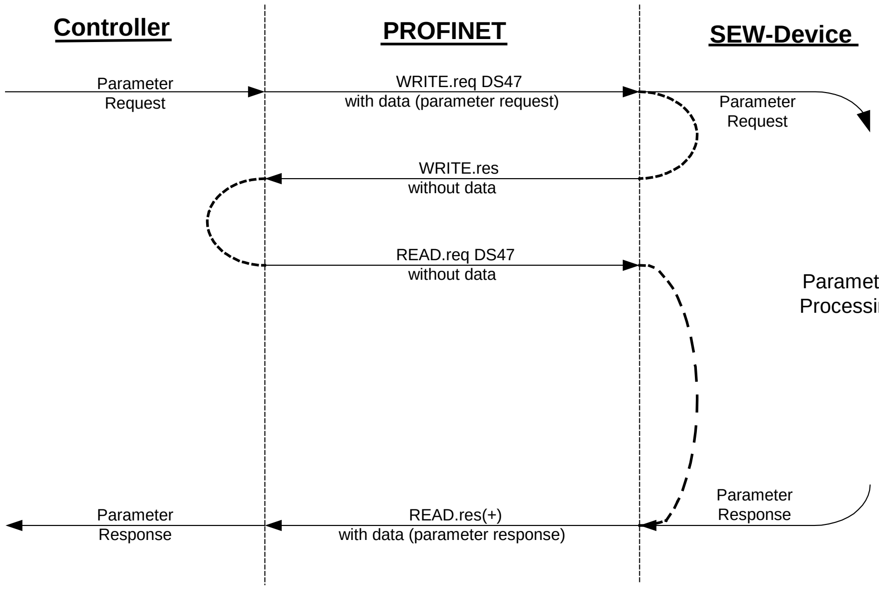
- 8-byte MOVILINK® parameter channel with all the services supported by the station such as
  - READ parameter
  - WRITE parameter
  - WRITE parameter volatile
  - etc.

Field	Data type	Values
	Unsigned8	0x00 Reserved 0x01 - 0xFF
Request ID	Unsigned8	**0x40 SEW MOVILINK® service** 0x41 SEW Data Transport
Response ID	Unsigned8	Response (+): 0x00 Reserved **0x40 SEW MOVILINK® service (+)** 0x41 SEW Data Transport Response (-): **0xC0 SEW MOVILINK® service (-)** 0x41 SEW Data Transport
	Unsigned8	0x00 - 0xFF Number of axes 0 - 255
No. of parameters	Unsigned8	0x01 - 0x13 1 - 19 DWORDs (240 DP-V1 data bytes)
Attribute	Unsigned8	**For SEW MOVILINK® (Request ID = 0x40):** **0x00 No service** **0x10 READ parameters** **0x20 WRITE parameter** **0x40 Read Minimum** **0x50 Read Maximum** **0x60 Read Default** **0x80 Read Attribute** **0x90 Read EEPROM** **0xA0 - 0xF0 Reserved** SEW data transport: 0x10 Value
No. of elements	Unsigned8	0x00 For parameters that are not indexed 0x01 - 0x75 Quantity 1 - 117
Parameter Number	Unsigned16	0x0000 - 0xFFFF MOVILINK® parameter index
Subindex	Unsigned16	0x0000
Format	Unsigned8	0x43 Double word 0x44 Error
No. of Values	Unsigned8	0x00 - 0xEA Quantity 0 - 234
Error Value	Unsigned16	0x0080 + MOVILINK®-Additional Code Low **For SEW MOVILINK® 16 Bit error value**

11.2.3 Parameterization procedure via data set 47

Parameter are accessed with the combination of the *WRITE RECORD* and *READ RECORD* PROFINET services. The parameter setting order is transferred to the IO device using the *WRITE.req*, Then it is processed internally.

The controller now sends a *READ.req* to pick up the parameter setting response. The device sends a positive response *READ.res*. The user data now contain the parameter setting response of the parameter setting order that was previously sent with *WRITE.req* (see the following figure). This mechanism applies to a PROFINET controller.

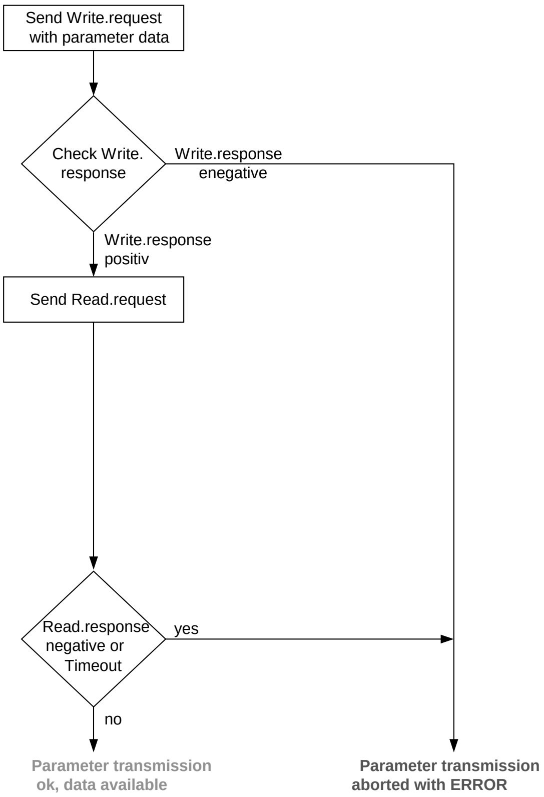


62208AXX

Figure 3: Telegram sequence for parameter access via Read/Write Record

11.2.4 Processing sequence for controller

If the bus cycles are very short, the request for the parameter response arrives before the SEW device has concluded the parameter access in the device. This means the response data from the SEW device is not available yet. In this state, the device delays the response to the *Read Record Request*.


Send Write.request with parameter data
Check Write. response
Write.response enegative
Write.response positiv
Send Read.request
Read.response negative or Timeout
yes
no
Parameter transmission ok, data available
Parameter transmission aborted with ERROR

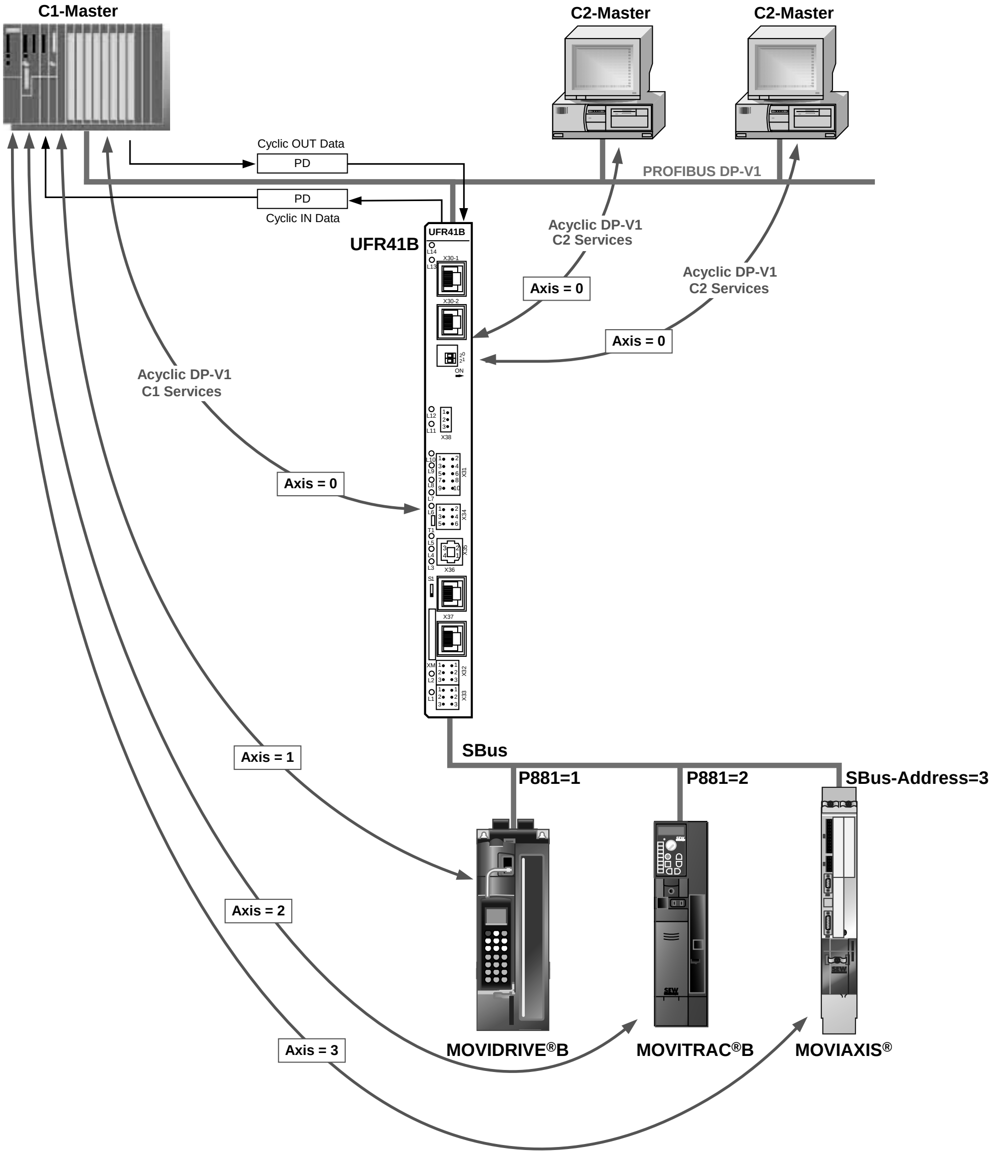
62209AEN

11.2.5 Addressing connected inverters

The structure of the DS47 data record defines an *axis* element. This element is used to reach multi-axis drives that are operated via one PROFINET interface. The *axis* element addresses one of the units connected via PROFINET interface.

Parameter access to lower-level stations

With the setting *Axis = 0*, the parameters of the UFR41B fieldbus gateway can be accessed directly. To being able to access slave units connected to the UFR41B fieldbus gateway, the setting must be *Axis = SBus address*. SBus address 15 must not be used when engineering via PROFIBUS or parameter services via PROFIBUS.



65203AXX

See the "Appendix" for a schematic representation of parameter access to lower-level units.

11.2.6 MOVILINK® parameter requests

The MOVILINK® parameter channel of the SEW inverter is directly mapped in the structure of data record 47. The Request ID 0x40 (SEW MOVILINK® service) is used for the exchange of MOVILINK® parameter setting requests. Parameter access with MOVILINK® services usually takes place according to the structure described below. The typical telegram sequence is used for data record 47.

Request ID: 0x40 SEW MOVILINK® **service**

The actual service is defined by the data set element *Attribute* in the MOVILINK® parameter channel. The high nibble of the element corresponds to the MOVILINK® service code.

Example for reading a parameter via MOVILINK®

The following tables give an example of the structure of the *WRITE.request* and *READ.response* user data for reading an individual parameter via the MOVILINK® parameter channel.

Sending a parameter request

The table shows the coding of the user data for the *WRITE.request* PROFINET service. The *WRITE.request* service is used to transfer the parameter setting request to the inverter. The firmware version is read.

The following table shows the WRITE request header for transferring the parameter request.

Service	WRITE. request	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	10	10 byte user data for parameter request

The following table shows the WRITE.request user data for MOVILINK® "Read parameters".

Byte	Field	Value	Description
0		0x01	Individual reference number for the parameter setting request is mirrored in the parameter response
1	Request ID	0x40	SEW MOVILINK® service
2	Axis	0x00	Axis number 0 = UFR41B 1 = Access to slave unit with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Attribute	0x10	MOVILINK® service "READ parameter"
5	No. of elements	0x00	0 = access to direct value, no subelement
6, 7	Parameter Number	0x206C	MOVILINK® index 8300 = "Firmware version"
8, 9	Subindex	0x0000	Subindex 0

Query parameter response

The following table shows the coding of the READ.request user data including the PROFINET header.

Service	READ. request	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	240	Maximum length of response buffer in the master

Positive MOVILINK® parameter setting response

The table shows the READ.response user data with the positive response data of the parameter setting request. The parameter value for index 8300 (firmware version of UFR41B) is returned as an example.

Service	READ. request	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	10	Maximum length of response buffer in the master

Byte	Field	Value	Description
0		0x01	Mirrored reference number from the parameter setting request
1	Response ID	0x40	Positive MOVILINK® response
2	Axis	0x00	0 = UFR41B
3	No. of parameters	0x01	1 parameter
4	Format	0x43	Parameter format: Double word
5	No. of values	0x01	1 value
6, 7	Value High	0x311C	Higher-order part of the parameter
8, 9	Value Low	0x7289	Lower-order part of the parameter
			Decoding: 0x 311C 7289 = 823947913 dec >> firmware version 823 947 9.13

Example for writing a parameter via MOVILINK®

The following tables show the an example of the structure of the *WRITE* and *READ* services for volatile writing of the value 12345 to the IPOS$^{plus®}$ variable H0 of a MOVIDRIVE® B with SBus address 1, which is connected to the CAN 1 system bus (X33) of UFR41B (parameter index 11000). The MOVILINK® service *WRITE Parameter volatile* is used for this purpose.

Send "WRITE parameter volatile" request

Service	WRITE. request	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	16	16-byte user data for order buffer

The following table shows the WRITE.request user data for MOVILINK® "Write parameters volatile.

Byte	Field	Value	Description
0		0x01	Individual reference number for the parameter setting request is mirrored in the parameter response
1	Request ID	0x40	SEW MOVILINK® service
2	Axis	0x01	1 = MOVIDRIVE® B with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Attribute	0x30	MOVILINK® service "WRITE parameter volatile"
5	No. of elements	0x00	0 = access to direct value, no subelement
6, 7	Parameter Number	0x2AF8	Parameter index 11000 = "IPOS variable H0"
8, 9	Subindex	0x0000	Subindex 0
10	Format	0x43	Double word
11	No. of values	0x01	Change 1 parameter value
12, 13	Value High word	0x0000	Higher-order part of the parameter value
14, 15	Value Low word	0x0BB8	Lower-order part of the parameter value

After sending this WRITE.request, the WRITE.response is received. If there was no status conflict in processing the parameter channel, a positive WRITE.response occurs. Otherwise, the status fault is located in Error_code_1.

Query parameter response

The following table shows the coding of the READ.req user data including the PROFINET header.

Service	READ. request	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	240	Maximum length of response buffer in the master

Positive response to "WRITE Parameter volatile"

Service	READ. response	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	4	4 byte user data in response buffer

Byte	Field	Value	Description
0		0x01	Mirrored reference number from the parameter setting request
1	Response ID	0x40	Positive MOVILINK® response
2	Axis	0x01	Reflected axis number 1 = MOVIDRIVE® B with SBus address 1
3	No. of parameters	0x01	1 parameter

Negative parameter response

The following table shows the coding of a negative response of a MOVILINK® service. Bit 7 is entered in the the response ID if the response is negative.

Service	WRITE. response	Description
API	0	Fixed setting = 0
Slot_Number	0	Random (is not evaluated)
Subslot_Number	1	Fixed setting = 1
Index	47	Index of the dataset for the parameter request; constant index 47
Length	8	8 byte user data in response buffer

Byte	Field	Value	Description
0		0x01	Mirrored reference number from the parameter setting request
1	Response ID	0xC0	Negative MOVILINK® response
2	Axis	0x01	1 = MOVIDRIVE® B with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Format	0x44	Error
5	No. of values	0x01	1 error code
6, 7	Error value	0x0811	MOVILINK® return code e.g. error class 0x08, Add. code 0x11 (see section "MOVILINK® configuration return codes for PROFINET" on page 140)

MOVILINK® return codes of PROFINET parameterization

The following table shows the return codes that are returned by the SEW PROFINET interface module in the event of an error in the PROFINET parameter access.

MOVILINK® return code (hex)	Description
0x0810	Invalid index, parameter index does not exist in the unit
0x0811	Function/parameter not implemented
0x0812	Read access only
0x0813	Parameter lock activated
0x0814	Factory setting is active
0x0815	Value for parameter too large
0x0816	Value for parameter too small
0x0817	Required option card not installed
0x0818	Error in system software
0x0819	Parameter access only via RS-485 process interface
0x081A	Parameter access only via RS-485 diagnostics interface
0x081B	Parameter is access-protected
0x081C	Controller inhibit is required
0x081D	Invalid value for parameter
0x081E	Factory setting was activated
0x081F	Parameter was not saved in EEPROM
0x0820	Parameter cannot be changed with output stage enabled / reserved
0x0821	Reserved
0x0822	Reserved
0x0823	Parameter may only be changed at IPOS program stop
0x0824	Parameter may only be changed when auto setup is deactivated
0x0505	Incorrect coding of management and reserved byte
0x0602	Communication error between inverter system and fieldbus interface
0x0502	Timeout of secondary connection (e.g. during reset or with Sys-Fault)
0x0608	Incorrect coding of the format field

11.2.7 PROFIdrive parameter orders

The PROFIdrive parameter channel of SEW inverters is directly mapped in the structure of data record 47. Parameter access with PROFIdrive services usually takes place according to the structure described below. The typical telegram sequence is used for data record 47. PROFIdrive only defines the two request IDs

Request ID: 0x01request parameter (PROFIdrive)

Request ID: 0x02change parameter (PROFIdrive)

This means data access is restricted in comparison with the MOVILINK® services.

TIP

The request ID *0x02 Change Parameter (PROFIdrive)* results in remanent write access to the selected parameter. Consequently, the internal flash/EEPROM of the inverter is written with each write access. Use the MOVILINK® service "WRITE Parameter volatile" if parameters must be written cyclically at short intervals. With this service, you only alter the parameter values in the RAM of the inverter.

Reading a parameter via PROFIdrive example

The following tables show an example of the structure of the WRITE.request and READ.res user data for reading an individual parameter via the MOVILINK® parameter channel.

Sending a parameter request

The table shows the coding of the user data for the WRITE.req service specifying the PROFINET header. The WRITE.req service is used to transfer the parameter setting request to the inverter.

Service:	WRITE.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	10	10 byte user data for parameter request

Byte	Field	Value	Description
0		0x01	Individual reference number for the parameter setting request is mirrored in the parameter response
1	Request ID	0x01	Request parameter (PROFIdrive)
2	Axis	0x00	0 = UFR41B 1 = Slave unit with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Attribute	0x10	Access to parameter value
5	No. of elements	0x00	0 = access to direct value, no subelement
6, 7	Parameter Number	0x206C	MOVILINK® index 8300 = "Firmware version"
8, 9	Subindex	0x0000	Subindex 0

Query parameter response

The following table shows the coding of the READ.req USER DATA including the PN header.

Service:	READ.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	240	Maximum length of response buffer in the PN controller

Positive PROFIdrive parameter response

The table shows the READ.res user data with the positive response data of the parameter setting request. The parameter value for index 8300 (firmware version) is returned as an example.

Service:	READ.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	10	10 byte user data in response buffer

Byte	Field	Value	Description
0		0x01	Mirrored reference number from the parameter setting request
1	Response ID	0x01	Positive response for "Request Parameter"
2	Axis	0x00	Reflected axis number 0 = UFR41B 1 = Slave units with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Format	0x43	Parameter format: Double word
5	No. of values	0x01	1 value
6, 7	Value Hi	0x311C	Higher-order part of the parameter
8, 9	Value Lo	0x7289	Lower-order part of the parameter
			Decoding: 0x 311C 7289 = 823947913 dec >> firmware version 823 947 9.13

Example for writing a parameter via PROFIdrive

The following tables show an example of the structure of the *WRITE* and *READ* services for the **remanent** writing of the internal setpoint n11 of a MOVIDRIVE® B with SBus address 1 that is connected to UFR41B via the CAN 1 system bus (see section "Example for writing a parameter via MOVILINK®" on page 138). The PROFIdrive *Change parameter* service is used for this purpose.

Send "WRITE parameter" request

The following table shows the PROFINET header of the WRITE request with parameter request.

Service:	WRITE.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	16	16-byte user data for order buffer

The following table shows the WRITE.req user data for the PROFINET service "Change Parameter".

Byte	Field	Value	Description
0		0x01	Individual reference number for the parameter setting request is mirrored in the parameter response
1	Request ID	0x02	Change parameter (PROFIdrive)
2	Axis	0x01	Axis number 1 = MOVIDRIVE® B with SBus address 1
3	No. of parameters	0x01	1 parameter
4	Attribute	0x10	Access to parameter value
5	No. of elements	0x00	0 = access to direct value, no subelement
6, 7	Parameter Number	0x2129	Parameter index 8489 = P160 n11
8, 9	Subindex	0x0000	Subindex 0
10	Format	0x43	Double word
11	No. of values	0x01	Change 1 parameter value
12, 13	Value HiWord	0x0000	Higher-order part of the parameter value
14, 15	Value LoWord	0x0BB8	Lower-order part of the parameter value

After sending this WRITE.request, the WRITE.response is received. If there was no status conflict in processing the parameter channel, a positive WRITE.response occurs. Otherwise, the status fault is located in Error_code_1.

Quering parameter response

The following table shows the coding of the WRITE.req user data including the PROFINET header.

Field	Value	Description
Function_Num		READ.req
Slot_Number	X	Slot_Number not used
Index	47	Index of the data record
Length	240	Maximum length of response buffer in the PN controller

Positive response to "WRITE Parameter"

The following table shows the PROFINET header of the positive READ.response with parameterization response.

Service:	READ.response	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	4	4 byte user data in response buffer

The following table shows the positive response for the PROFINET service "Change Parameter".

Byte	Field	Value	Description
0		0x01	Mirrored reference number from the parameter setting request
1	Response ID	0x02	Positive PROFIdrive response
2	Axis	0x01	1 = SBus address 1 of MOVIDRIVE® B
3	No. of parameters	0x01	1 parameter

Negative parameter response

The following table shows the coding of a negative response of a PROFIdrive service. Bit 7 is entered in the response ID if the response is negative.

Service:	READ.response	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data record; constant index 47
Length	8	8 byte user data in response buffer

Byte	Field	Value	Description
0	Response reference	0x01	Mirrored reference number from the parameter setting request
1	Response ID	0x810x82	Negative response for "Request Parameter" Negative response for "Change Parameter"
2	Axis	0x01	1 = SBus address 1 of MOVIDRIVE® B
3	No. of parameters	0x01	1 parameter
4	Format	0x44	Error
5	No. of values	0x01	1 error code
6, 7	Error value	0x0811	MOVILINK® return code e.g. error class 0x08, Add. code 0x11 (see section "MOVILINK® configuration return codes for PROFINET" on page 140)

PROFIdrive return codes for PROFINET

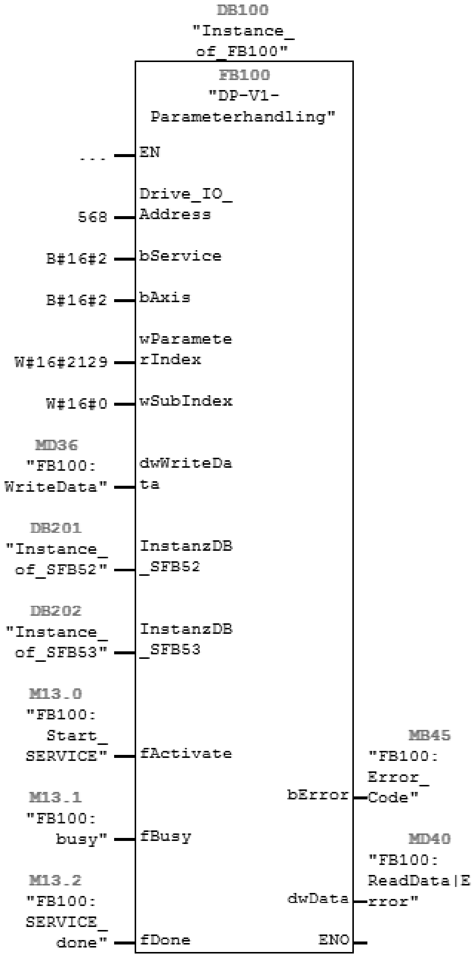
The following table shows the coding of the error number in the PROFIdrive parameter response according to PROFIdrive profile V3.1. This table applies if you use the PROFIdrive services "Request parameter" and/or "Change parameter".

Error no.	Meaning	Used for
0x00	Invalid parameter number	Access to non-existent parameters
0x01	Parameter value cannot be changed	An attempt was made to change a parameter value that cannot be changed
0x02	Minimum or maximum value exceeded	An attempt was made to change a value to one that is outside of the limit values
0x03	Incorrect subindex	Access to non-existent subindex
0x04	No assignment	Access with subindex to parameter that is not indexed
0x05	Incorrect data type	An attempt was made to change a replace a value with one that does not correspond to the data type of the parameter
0x06	Setting not permitted (can only be reset)	An attempt was made to set a value to one larger than 0 where this is not permitted
0x07	Description element cannot be changed	Access to description element that cannot be changed
0x08	Reserved	(PROFIdrive Profile V2: PPO write request for IR not available)
0x09	Description does not exist	Access to description that is not accessible (parameter value exists)
0x0A	Reserved	(PROFIdrive Profile V2: incorrect access group)
0x0B	No operation priority	An attempt was made to change a parameter without change rights
0x0C	Reserved	(PROFIdrive Profile V2: incorrect password)
0x0D	Reserved	(PROFIdrive Profile V2: text cannot be read in cyclic data transfer)
0x0E	Reserved	(PROFIdrive Profile V2: name cannot be read in cyclic data transfer)
0x0F	No text assignment available	Access to text assignment that is not accessible (parameter value exists)
0x10	Reserved	(PROFIdrive Profile V2: no PPO write)
0x11	Request cannot be executed due to the operating mode	Access is currently not possible and the reason is not explained
0x12	Reserved	(PROFIdrive Profile V2: other error)
0x13	Reserved	(PROFIdrive Profile V2: data cannot be read in cyclic exchange)
0x14	Incorrect value	An attempt was made to change a value to one that is in the permitted range but is not permitted due to other long-term reasons (parameter with specified individual values)
0x15	Response is too long	The length of the current response exceeds the maximum transmittable length
0x16	Invalid parameter address	Invalid value or value that is not valid for this attribute, number of elements, parameter number, subindex or a combination of these factors.
0x17	Incorrect format	Write request: Invalid format or parameter data format that is not supported
0x18	Number of values is not consistent	Write request: Number of values of parameter data does not correspond to the number of elements in the parameter address
0x19	Axis does not exist	Access to an axis that does not exist
up to 0x64	Reserved	-
0x65..0xFF	Depends on the manufacturer	-

11.2.8 Example program for SIMATIC S7

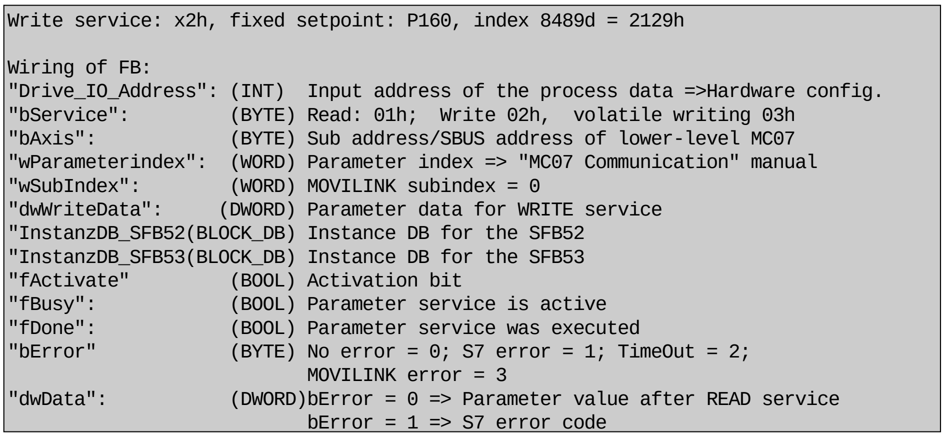
	TIPS
	The MOVILINK® parameter channel example program is available from the SEW homepage under "Software". This example is a special and free service that demonstrates only the basic approach to generating a PLC program. SEW is not liable for the contents of the sample program.

- Calling the function module:



12101AXX

- Comment regarding the function module:



```
Write service: x2h, fixed setpoint: P160, index 8489d = 2129h

Wiring of FB:
"Drive_IO_Address": (INT)  Input address of the process data =>Hardware config.
"bService":         (BYTE) Read: 01h;  Write 02h,  volatile writing 03h
"bAxis":            (BYTE) Sub address/SBUS address of lower-level MC07
"wParameterindex":  (WORD) Parameter index => "MC07 Communication" manual
"wSubIndex":        (WORD) MOVILINK subindex = 0
"dwWriteData":     (DWORD) Parameter data for WRITE service
"InstanzDB_SFB52(BLOCK_DB) Instance DB for the SFB52
"InstanzDB_SFB53(BLOCK_DB) Instance DB for the SFB53
"fActivate"         (BOOL) Activation bit
"fBusy":            (BOOL) Parameter service is active
"fDone":            (BOOL) Parameter service was executed
"bError"            (BYTE) No error = 0; S7 error = 1; TimeOut = 2;
                           MOVILINK error = 3
"dwData":           (DWORD)bError = 0 => Parameter value after READ service
                           bError = 1 => S7 error code
```

65377AEN

12 Operating MOVITOOLS® MotionStudio

12.1 About MOVITOOLS® MotionStudio

12.1.1 Tasks

The MOVITOOLS® MotionStudio software package enables you to perform the following tasks:

- Establishing communication with units
- Executing functions with the units

12.1.2 Establishing communication with units

The SEW Communication Server is integrated into MOVITOOLS® MotionStudio for establishing communication with the units.

The SEW Communication Server allows you to create **communication channels**. Once the channels are established, the units communicate via these communication channels using their communication options. You can operate up to four communication channels at the same time.

Depending on the unit and its communication options, the following communication channels are available:

- Serial (RS485) via interface adapters
- System bus (SBus) via interface adapters
- Ethernet
- EtherCAT
- Fieldbus
- PROFIBUS DP/DP-V1
- S7-MPI

12.1.3 Executing functions with the units

MOVITOOLS® MotionStudio enables you to perform the following functions:

- Parameterization (for example in the parameter tree of the unit)
- Startup
- Visualization and diagnostics
- Programming

The following basic components are integrated into MOVITOOLS® MotionStudio allowing you to use the units to execute functions:

- MotionStudio
- MOVITOOLS®

All functions communicate using **tools**. MOVITOOLS® MotionStudio provides the right tools for every unit type.

12.2 First steps

12.2.1 Starting the software and creating a project

Proceed as follows to start MOVITOOLS® MotionStudio and create a project:

1. Start MOVITOOLS® MotionStudio in the WINDOWS® start menu via the following path:

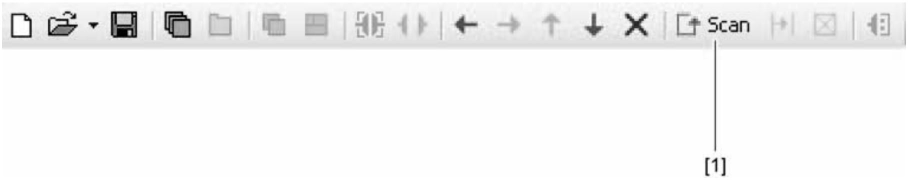
   "Start\Program\SEW\MOVITOOLS MotionStudio\MOVITOOLS MotionStudio"
2. Create a project with name and storage location.

12.2.2 Establishing communication and scanning the network

Proceed as follows to establish a communication with MOVITOOLS® MotionStudio and scan your network:

1. Set up a communication channel to communicate with your units.

   For detailed information on how to configure a communication channel, see the section regarding the relevant communication type.
2. Scan your network (unit scan). To do so, click the [Start network scan] button [1] in the toolbar.



64334AXX

3. Select the unit you want to configure.
4. Open the context menu with a right mouse click.

   As a result you will see a number of unit-specific tools to execute various functions with the units.

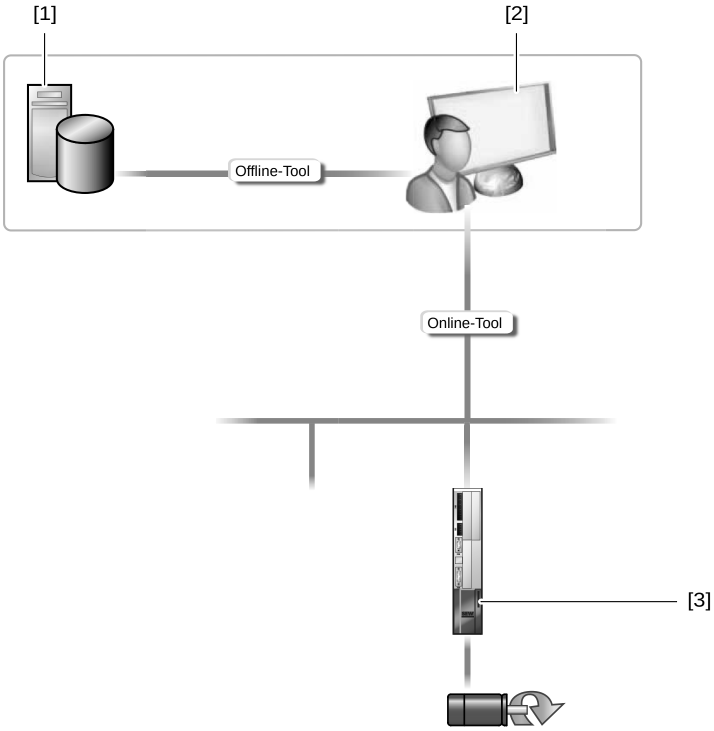
12.3 Communication mode

12.3.1 Overview

MOVITOOLS® MotionStudio differentiates between "online" and "offline" communication mode.

You can select the communication mode. Unit-specific offline or online tools are provided depending on the communication mode you have selected.

The following figure illustrates the two types of tools:



64335AXX

Tools	Description
Offline tools	Changes made using offline tools affect **"ONLY"** the RAM [2]. • Save your project so that the changes can be stored on the hard disk [1] of your PC. • To transfer the changes also to your unit [3], perform a download.
Online tools	Changes made using online tools affect **"ONLY"** the unit [3]. • To transfer the changes to the RAM [2], perform an upload. • Save your project so that the changes can be stored on the hard disk [1] of your PC.

TIP

The "online" communication mode is "NOT" a response message which informs you that you are currently connected to the unit or that your unit is ready for communication.

- Should you require this feedback, observe section "Setting the cyclical accessibility test" in the online help (or the manual) of MOVITOOLS® MotionStudio.

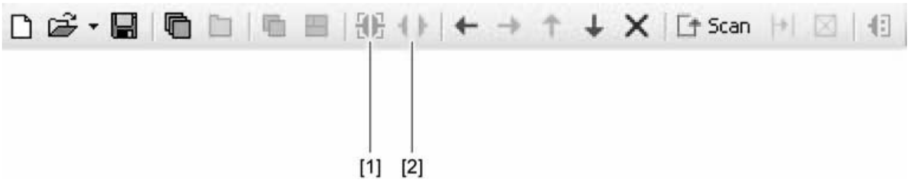
TIP

- Project management commands (such as "download" and "upload"), the online unit status, and the "unit scan" operate independently of the set communication mode.
- MOVITOOLS® MotionStudio starts up in the communication mode that you set before you closed the program.

12.3.2 Selecting communication mode (online or offline)

Proceed as follows to select a communication mode:

1. Select the communication mode:
   - "Online" [1] for functions (online tools) that should directly influence the unit.
   - "Offline" [2] for functions (offline tools) that should influence your project.



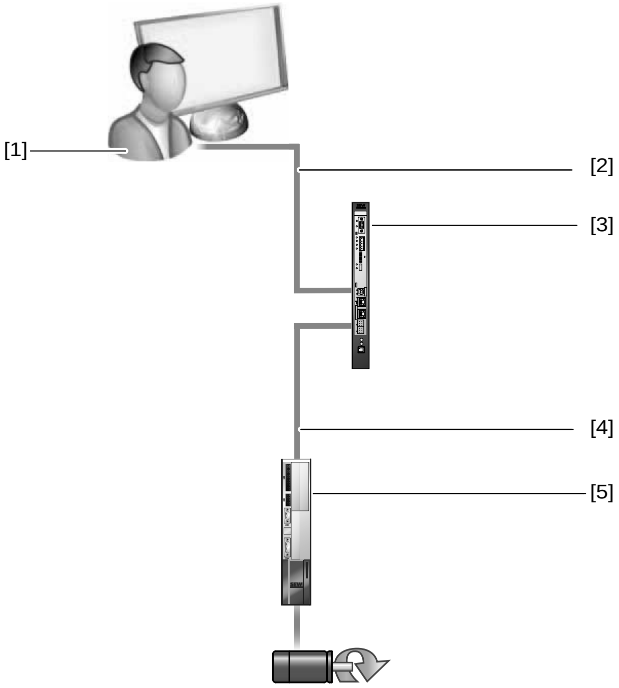
64337AXX

2. Select the unit node.
3. Right-click to open the context menu and display the tools for configuring the unit.

12.4 Communication via USB (direct)

12.4.1 Connect the unit with the PC using USB connection cables

The illustration shows how the unit (in the example a fieldbus gateway [3]) is connected with the PC [1] using a USB connection cable [2]. It also shows how the fieldbus gateway [3] is connected with the lower-level unit [5] via SBus (CAN).



64782AXX

[1] PC with USB interface

[2] USB connection cable

[3] Fieldbus gateway (UFx41 for example)

[4] SBus connection (CAN based) between fieldbus gateway and lower-level unit

[5] Lower-level unit (MOVIAXIS® for example)

Do the following to connect the UFx41B fieldbus gateway with the PC and the lower-level units:

1. Insert the **A** connector of the USB cable [2] into a free USB port on your PC [1].
2. Insert the **B** connector of the USB cable [2] into the USB port on your fieldbus gateway [3].
3. Connect the SBus interface of the fieldbus gateway [3] with the SBus interface of the lower-level unit [5].

12.4.2 Installing the drivers

Before you can communicate with the unit via USB (direct), you have to install the required driver file from the installation path of MOVITOOLS® MotionStudio.

Follow the instructions below to install the driver for USB communication:

1. Connect the unit to a free USB port on your PC.

   Your PC will detect the new hardware and launch the hardware wizard.
2. Follow the instructions of the hardware wizard.
3. Click on [Browse] and go to the MOVITOOLS® MotionStudio installation folder.
4. Enter the following path:

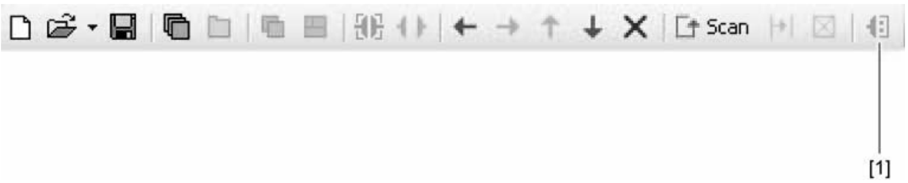
   "..\Program Files\SEW\MotionStudo\Driver\SEW_USBWIN32_051120"
5. Click [Next] to install the driver.

12.4.3 Configuring USB communication

You need a USB connection between your PC and the units you want to configure.

Proceed as follows to configure USB communication:

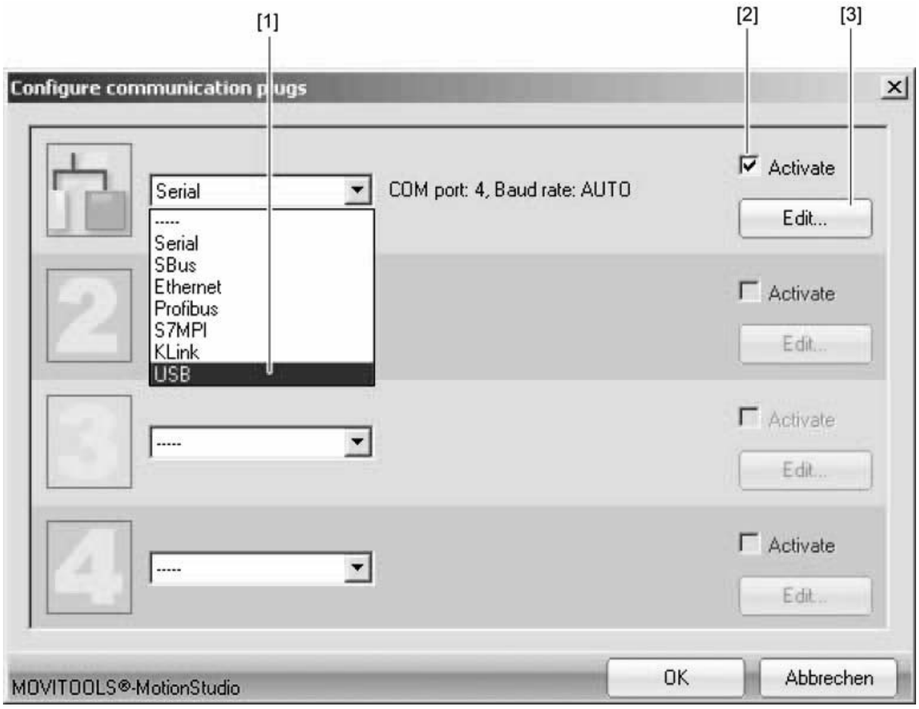
1. Click on "Configure communication plugs" [1] in the toolbar.



64341AXX

[1] "Configure communication plugs" icon

Doing so will open the "Configure communication plugs" window.



COM port: 4, Baud rate: AUTO

64743AEN

[1] "Communication type" selection field

[2] "Activated" checkbox

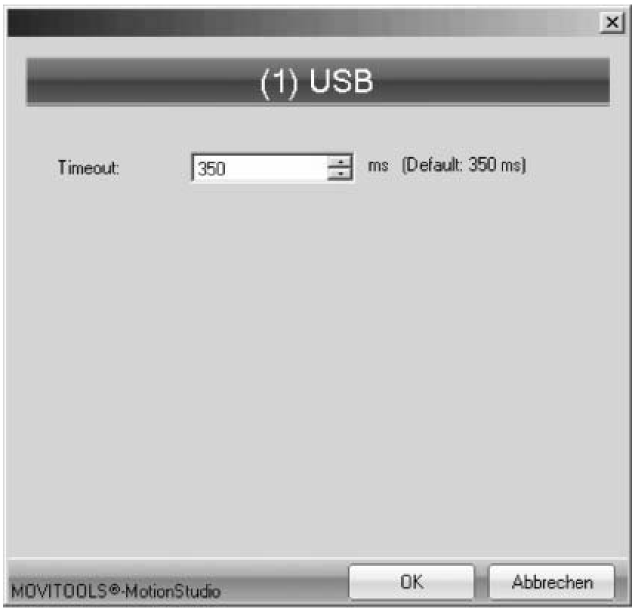
[3] [Edit] button

2. From selection field [1], choose the communication type "USB (direct)".

   In the example, "USB" is activated as the communication type for the first communication channel [2].

3. Press the [Edit] button [3] on the right side of the "Configure communication plugs" window.

This will display the settings for the "USB" communication type.



Timeout: 350 ms (Default: 350 ms)

12110AEN

4. Change the set communication parameters if necessary. When doing so, refer to the detailed description of the communication parameters.

12.4.4 USB communication parameters

The following table describes the communication parameters for the USB communication channel:

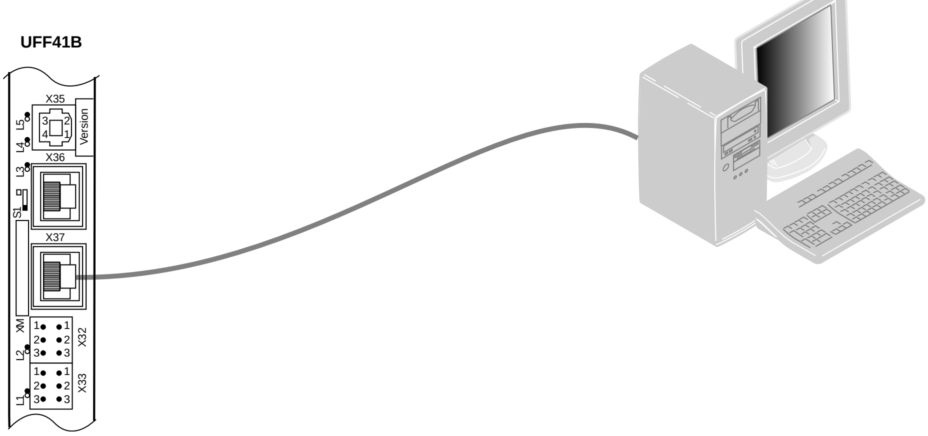
Communication parameters	Description	Note
Timeout	Waiting time in milliseconds that the master waits for a response from a slave after it has sent a request.	Default setting: 350 ms

12.5 Communication via Ethernet

12.5.1 Connecting the unit with the PC via Ethernet

Connecting the Ethernet interface of the UFx41B to the PC

The following figure shows the connection between the PC/laptop and the UFx41B.



64420AXX

The UFx41B can be connected either directly to the PC or via an Ethernet network. The Ethernet interfaces support auto crossing and auto negotiation for baud rate and duplex mode.

TIP

- The IP parameters of the UFx41B for X37 have to be as described in chapter "Setting the IP address parameters".
- "IP parameters X30-1 and X30-2 can also be set using the Address Editor (see chapter "Address Editor").

12.5.2 Address Editor

Overview

Address Editor is a free software tool from SEW-EURODRIVE. It is available once MOVITOOLS® MotionStudio has been installed.

You can use Address Editor to establish communication with your units via Ethernet and to address the units.

If you connect the Ethernet interface of your engineering PC to the Ethernet using a patch cable, Address Editor detects all Ethernet stations in the connected network segment (local power supply).

Unlike with "MOVITOOLS® MotionStudio", you will **not** have to adjust the IP address of the engineering PC to the local network.

This means the Address Editor is a useful addition to "MOVITOOLS® MotionStudio".

Proceed as follows if you have added other Ethernet stations to an existing network:

- Start the Address Editor
- Search for Ethernet stations

Once you have located the added Ethernet stations, continue using one of the following options:

- Adjust detected Ethernet stations to the network (address)
- Set the engineering PC appropriately for the network

Starting the Address Editor

You can use the Address Editor once MOVITOOLS® MotionStudio has been installed.

Do the following to start Address Editor:

1. Close MOVITOOLS® MotionStudio.
2. Start the Address Editor in the WINDOWS start menu by following the path below: "Start\Program\SEW\MOVITOOLS MotionStudio\Address Editor (Address Tool)"

Searching for Ethernet stations

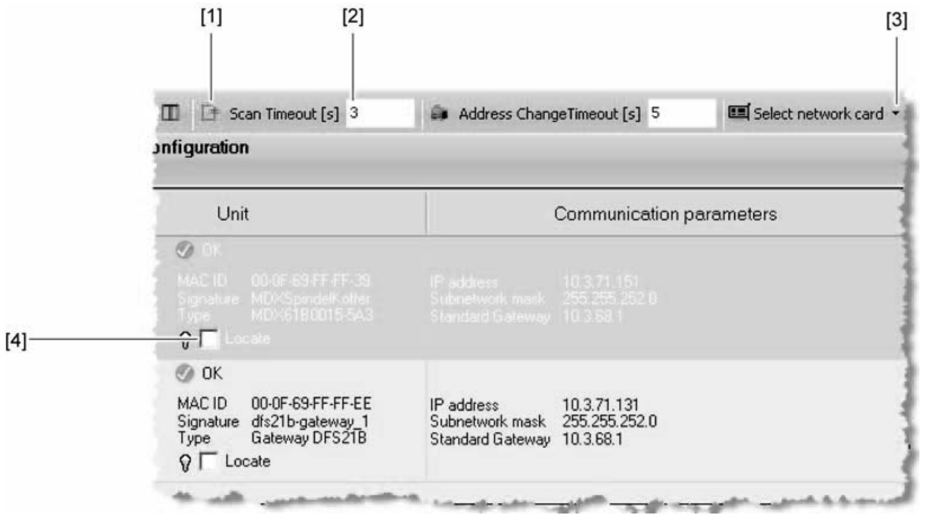
You can use the Address Editor to search a network for Ethernet stations. You can also specifically search recently added Ethernet stations. The Address Editor also helps you to locate the Ethernet interface of found Ethernet stations.

To search for Ethernet stations and locate hardware, proceed as follows:

1. Select "Ethernet" as the interface for PC and unit. To do so, click on the corresponding option field in the lower part of the window.
2. Click [Next] to confirm your selection and proceed to the next dialog.
3. Wait for the network scan to start **automatically**. The default setting for the waiting time (scan timeout) is 3 seconds [2]

   You can also start the network scan **manually** as follows:

   - If you have several network cards installed in your PC, select the required card. To do so, click the "Select network card" symbol [3] in the toolbar.
   - Click the "Start network scan" symbol [1] in the toolbar.



Scan Timeout [s] 3
Address ChangeTimeout [s] 5

Unit	Communication parameters
OK MAC ID 00-0F-69-FF-FF-39 Signature MDXSpindelKolter Type MDX61B0015-5A3 □ Locate	IP address 10.3.71.151 Subnetwork mask 255.255.252.0 Standard Gateway 10.3.68.1
OK MAC ID 00-0F-69-FF-FF-EE Signature dfs21b-gateway_1 Type Gateway DFS21B □ Locate	IP address 10.3.71.131 Subnetwork mask 255.255.252.0 Standard Gateway 10.3.68.1

64348AXX

[1] "Start network scan" symbol

[2] "Scan timeout" edit box

[3] "Select network card" symbol

[4] "Locate" checkbox

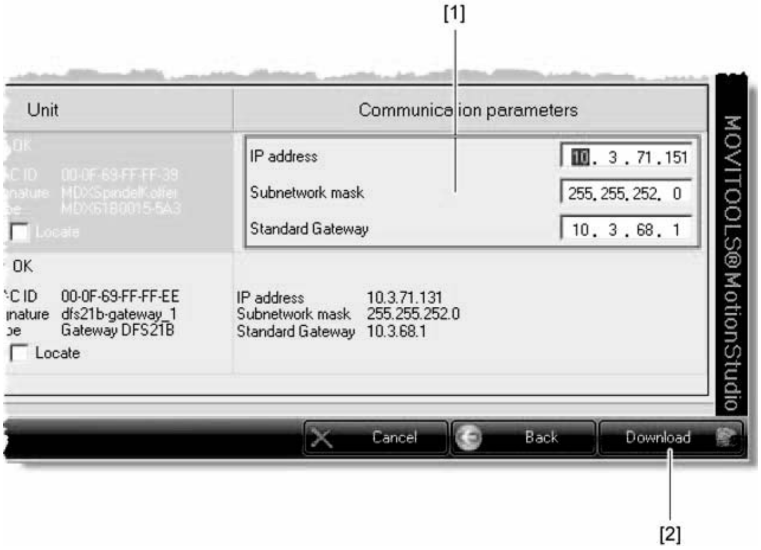
As a result, the current addresses of all Ethernet stations in the connected network will be displayed.

4. Mark the "Locate" checkbox [4] to locate an Ethernet station. As a result, the Link/Activity LED of the first Ethernet interface of the relevant Ethernet station will flash.

Adjust detected Ethernet stations to the network (address)

To set the located Ethernet stations appropriately for the network (addressing), proceed as follows:

1. Double-click on the "Communication parameters" window of the respective unit [1] to adjust the IP parameters of an Ethernet station to the network.



Unit	Communication parameters
OK :C ID 00-0F-69-FF-FF-39 nature MDXSpindelKoller be MDX61B0015-5A3 □ Locate	IP address 10. 3 . 71 . 151 Subnetwork mask 255. 255. 252. 0 Standard Gateway 10. 3 . 68. 1
OK :C ID 00-0F-69-FF-FF-EE nature dfs21b-gateway_1 be Gateway DFS21B □ Locate	IP address 10.3.71.131 Subnetwork mask 255.255.252.0 Standard Gateway 10.3.68.1

64349AXX

[1] "Communication parameters" window range

[2] "Download" button

The following fields can then be edited:

- IP address of the Ethernet station
- IP address of the subnetwork mask
- IP address of the standard gateway
- DHCP startup configuration (if supported by the unit)

2. Transmit the address changes to the Ethernet station. Click on [Download] [2].
3. Switch the unit off and back on again so the modified settings become effective.

Adjusting the engineering PC to the network (address)

To set the engineering PC appropriately for the network (addressing), proceed as follows:

1. Under [Start] / [Settings] / [Network and Dial-up Connections], select the PC interface you require.
2. Select "Properties" from the context menu.
3. Activate the checkbox by entering "Internet protocol (TCP/IP)".
4. Click on the "Properties" button.
5. For the subnetwork mask and standard gateway, enter the same IP addresses that are used for other Ethernet stations in this local network.
6. For the engineering PC, enter an IP address that meets the following conditions:
   - In the blocks that define the **network**, the address section of the engineering PC must correspond with the address section of the other Ethernet stations.
   - In the blocks that define the **station**, the address section of the engineering PC must be different from the address section of the other Ethernet stations.
   - Do not assign the values "0", "4", "127" and "255" in the last block.

TIP

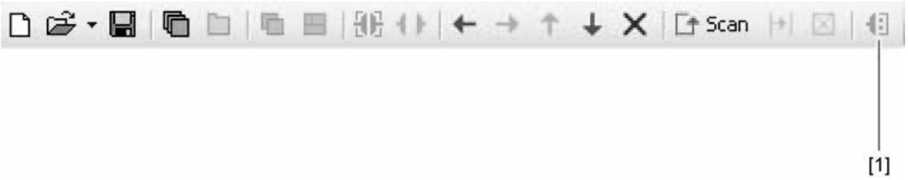
In the IP address of the subnetwork mask (e.g. 255.255.255.0), the values in the blocks have the following meaning:

- "255" defines the address of the network where the stations are located.
- "0" defines the address of the actual station to differentiate it from the others.

12.5.3 Configuring the communication channel via Ethernet

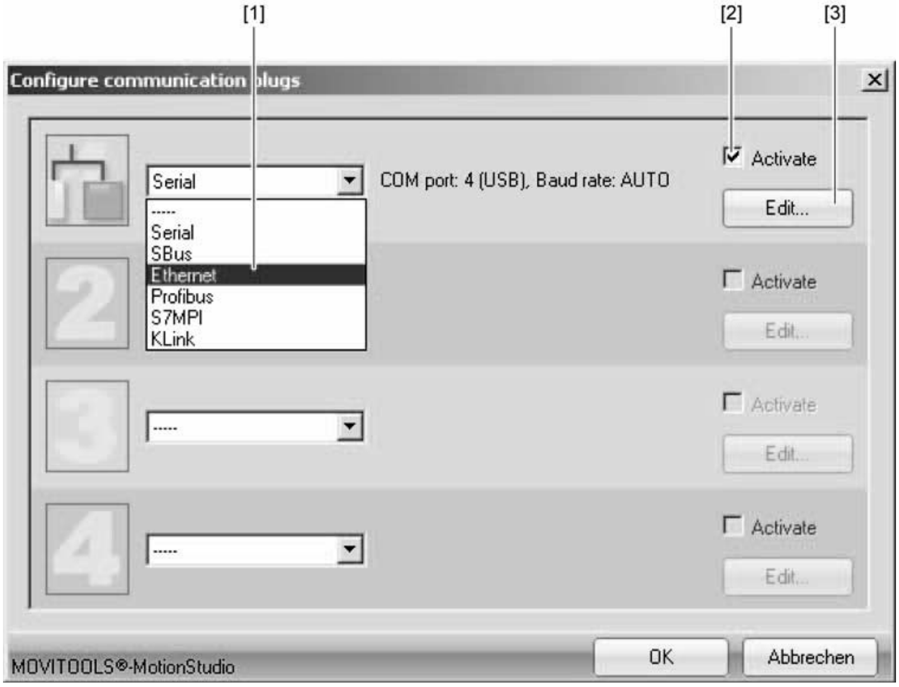
Proceed as follows to configure a communication channel for Ethernet:

1. Click on [Configure communication plugs] [1] in the toolbar.



64341AXX

2. This opens the "Configure communication plugs" window. From the list [1], select "Ethernet" as the communication type. In the example, "Ethernet" is activated as the communication type for the first communication channel [2].



64351AEN

3. Press the [Edit] button [3] in the right section of the window. This displays the settings for the "Ethernet" communication type.
4. Set up the SMLP protocol. To do so, select the "SMLP settings" tab.
5. Set the parameters. Follow the instructions described in the section "Setting parameters for SMLP".

	TIP
i	**SMLP** stands for **S**imple **M**OVI**L**INK® **P**rotocol. It is the unit protocol from SEW-EURODRIVE.

12.5.4 Setting communication parameters for SMLP

SMLP communication parameters

The following table describes the communication parameters for SMLP:

Communication parameters of the simple MOVILINK® protocol	**Description**	**Note**
Timeout	Waiting time in [ms] that the client waits for a response from the server after it has made a request.	• Default setting: 1000 ms • Increase the value as required if a delay in communication is causing malfunctions.
Broadcast IP address	IP address of the local network segment **within** which the unit scan is carried out.	In the default setting, the unit scan only detects units that are in the local network segment.
IP address of SMLP server	IP address of the SMLP server or of other units that are to be included in the unit scan but are **outside** the local network segment.	• Enter the IP address of units that are to be included in the unit scan but are **outside** the local network segment.
Excluded IP address	IP addresses of units that should **not** be included in the unit scan.	Enter the IP address of units that should **not** be included in the unit scan. This can be units that are not ready for communication (for example because they have not been started up yet).

To set up communication parameters for communicating via Ethernet, proceed as follows:

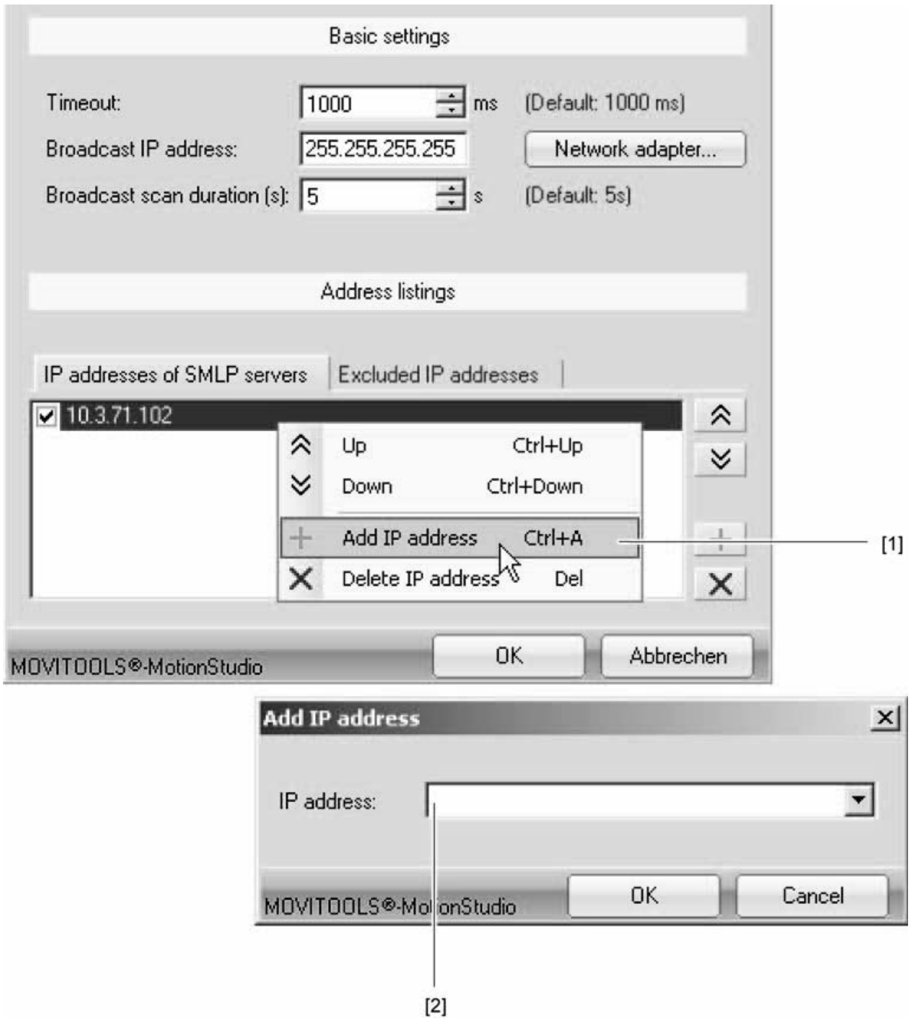
1. If necessary, change the preset communication parameters. Refer to the detailed description of the communication parameters for SMLP.

TIP

During a unit scan, the system recognizes only units that are in the same (local) network segment as the PC that is running on MOVITOOLS® MotionStudio.

- If you have units that are **OUTSIDE** the local network segment, add the IP addresses of these units to the list of SMLP servers.

2. To add an IP address, open the context menu and select [Add IP address] [1].


Basic settings

Timeout: 1000 ms (Default: 1000 ms)

Broadcast IP address: 255.255.255.255

Broadcast scan duration (s): 5 s (Default: 5s)

Address listings

IP addresses of SMLP servers | Excluded IP addresses

☑ 10.3.71.102

64352AEN

3. Add the IP address [2]

12.6 *Executing functions with the units*

12.6.1 Parameterizing units in the parameter tree

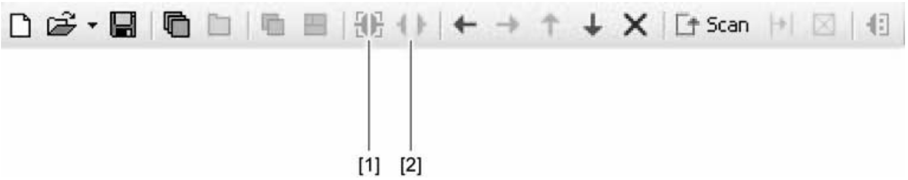
The parameter tree displays all unit parameters, grouped in folders.

You can manage unit parameters with the context menu or the toolbar. The following chapter describes how to read or change unit parameters.

12.6.2 Reading/changing unit parameters

To read or change unit parameters, proceed as follows:

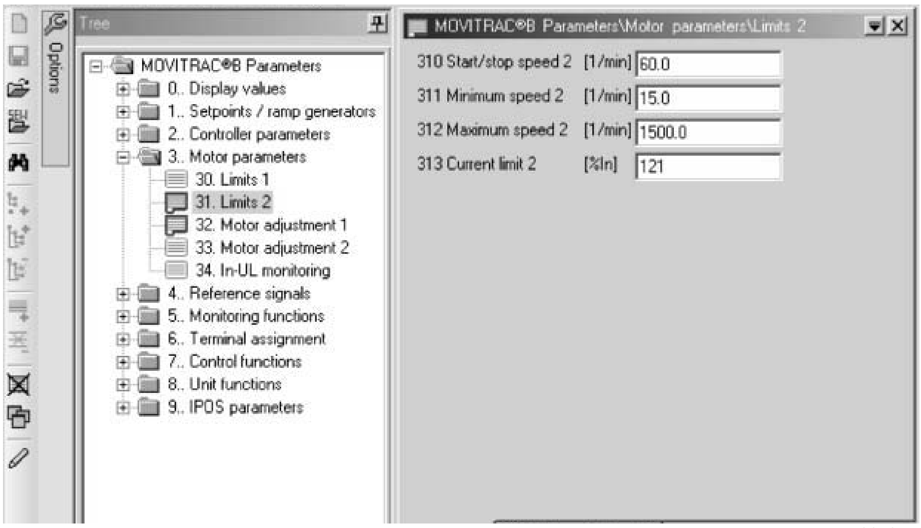
1. Switch to the required view (project view or network view).
2. Select the communication mode:
   - Click the [Switch to online mode] button [1] if you want to read or change parameters directly on the **unit**.
   - Click the [Switch to offline mode] button [2] if you want to read or change parameters in the **project**.



64337AXX

3. Select the unit you want to set parameters for.
4. Open the context menu and select the [Parameter tree] command.

   This opens the "Parameter tree" view on the right section of the screen.
5. Expand the "Parameter tree" up to the node you require.



- MOVITRAC®B Parameters
  - 0.. Display values
  - 1.. Setpoints / ramp generators
  - 2.. Controller parameters
  - 3.. Motor parameters
    - 30. Limits 1
    - 31. Limits 2
    - 32. Motor adjustment 1
    - 33. Motor adjustment 2
    - 34. In-UL monitoring
  - 4.. Reference signals
  - 5.. Monitoring functions
  - 6.. Terminal assignment
  - 7.. Control functions
  - 8.. Unit functions
  - 9.. IPOS parameters

MOVITRAC®B Parameters\Motor parameters\Limits 2

310 Start/stop speed 2 [1/min] 60.0
311 Minimum speed 2 [1/min] 15.0
312 Maximum speed 2 [1/min] 1500.0
313 Current limit 2 [%In] 121

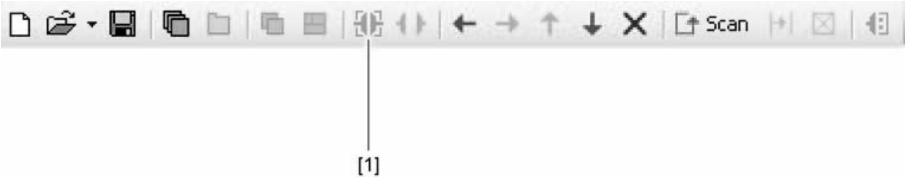
12079AEN

6. Double-click to display a particular group of unit parameters.
7. Press the enter key to finalize any changes you make to numerical values in the input fields.

12.6.3 Starting up the units (online)

To startup units (online), proceed as follows:

1. Switch to the network view.
2. Click the [Switch to online mode] button [1].



64354AXX

3. Select the unit you want to startup.
4. Open the context menu and select the command [Diagnostics] / [UFx Gateway Configurator].

   The Gateway Configurator opens.

	TIPS
i	• For detailed information about the unit parameters, refer to parameter list for the unit. • For detailed information about how to use the startup wizard, refer to the MOVITOOLS® MotionStudio online help.

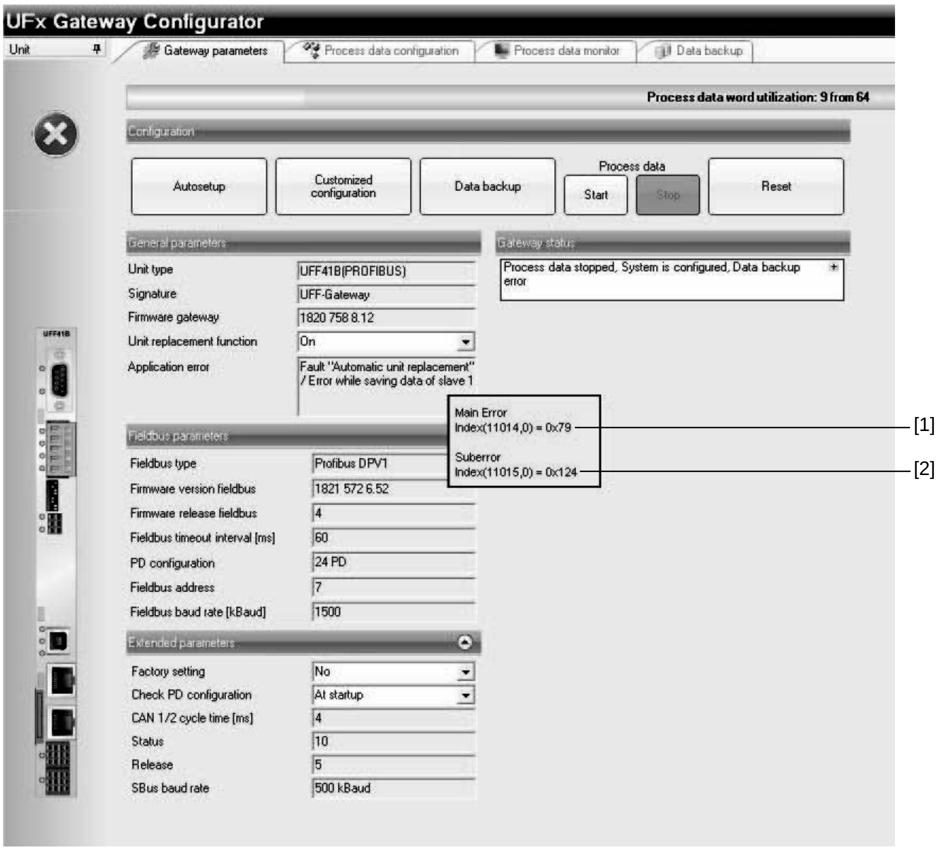
12.7 Special configuration and diagnostics tools

To configure the UFR41B in gateway operation, you can use the context menu to start both the "UFx gateway configurator" and the parameter tree. In addition to the configuration, it provides information for diagnostics of the gateway operation and displays transmitted process data (see chapter "Configuring the UFx41B fieldbus gateway and inverters").

13 Error Diagnostics

13.1 Error messages of the fieldbus gateway

Error messages of the fieldbus gateway are displayed in MOVITOOLS® MotionStudio via the "UFx Gateway Configurator" tool (Gateway parameter tab). The fieldbus gateway diagnoses an error number [1] with the associated suberror number [2]. In the following tables, this suberror number [2] is given in hexadecimal notation. It can be used to generate a suberror code referring to the relevant slave unit (see figure below).



Process data word utilization: 9 from 64

General parameters

Unit type	UFF41B(PROFIBUS)
Signature	UFF-Gateway
Firmware gateway	1820 758 8.12
Unit replacement function	On
Application error	Fault "Automatic unit replacement" / Error while saving data of slave 1

Gateway status

Process data stopped, System is configured, Data backup error

Main Error
Index(11014,0) = 0x79

Suberror
Index(11015,0) = 0x124

Fieldbus parameters

Fieldbus type	Profibus DPV1
Firmware version fieldbus	1821 572 6.52
Firmware release fieldbus	4
Fieldbus timeout interval [ms]	60
PD configuration	24 PD
Fieldbus address	7
Fieldbus baud rate [kBaud]	1500

Extended parameters

Factory setting	No
Check PD configuration	At startup
CAN 1/2 cycle time [ms]	4
Status	10
Release	5
SBus baud rate	500 kBaud

64881AEN

13.1.1 General errors of the fieldbus gateway

Error			
Description	**Number (hex)**	**Response**	**Remedy**
Wrong unit as fieldbus gateway. The SD card of the fieldbus gateway was instered in a MOVI-PLC® *advanced* DHF41B or DHR41B.	239.0	Fieldbus gateway remains in "Initialization" state.	Use the SD card of the fieldbus gateway only with UFx41B fieldbus gateway.
Error during communication between gateway program and gateway hardware. The error occurs when starting the unit.	239.1		Use the SD card of the fieldbus gateway only with UFx41B fieldbus gateway. Contact SEW service.
Error while scanning the slave units. Error while reading the unit type of the slave unit.	239.[No. of the slave unit]01	Fieldbus gateway initializes completely and starts process data communication with the other slave units.	The SBus addresses of various unit types were changed. Check for correct addressing of all slave units.
Error while scanning the slave units. Configured slave unit not found	111. [No. of the slave unit]02		Check whether all slave units are switched on when starting the fieldbus gateway and whether they have the correct SBus address. Check the SBus installation and the terminating resistors.

13.1.2 Error during process data processing

Error			
Description	**Number (hex)**	**Response**	**Remedy**
Master/slave configuration error: The total of configured process data of the slave units is higher than the fieldbus process image.	238.10100	Process data are not started.	The fieldbus process data configuration has to be expanded.
Master/slave configuration error: Process data length of slave unit too short	238.[No. of the slave unit]10		Configure the slave units with a minimum number of one PD per slave unit.
Master/slave configuration error: Process data length of slave unit too short	238.[No. of the slave unit]11		Check the following limit values of the process data configuration: • MOVIAXIS ®: Max. 16 PDs • MOVIDRIVE® B: Max. 10 PDs • MOVITRAC® B: Max. 3 PDs
Error while scanning the slave units: Configured slave unit not found	111.[No. of the slave unit]02	Fieldbus gateway initializes completely and starts process data communication with the other slave units.	Check whether all slave units are switched on when starting the fieldbus gateway and whether they have the correct SBus address. Check the SBus installation and the terminating resistors.
Error while stopping/starting the process data of the MOVIAXIS® slave unit	239.[No. of the slave unit]12	This slave does not contain any process data.	Check whether the MOVIAXIS® parameter setting level is set to "planning engineer".
Error during automatic of the MOVIAXIS® PDO configuration: Error while setting the parameters for the process data of the MOVIAXIS® slave unit.	238.[No. of the slave unit]13		Check whether the MOVIAXIS® parameter setting level is set to "planning engineer". Check whether process data objects (PDOs) in the axis modules were configured with CAN IDs which are needed by the gateway for communication.
Error during PDO configuration of the MOVIAXIS® slave unit	238.[No. of the slave unit]14		
Internal system error in the fieldbus gateway	239.10600 239.10610 239.10620 239.10630	Process data are neither initialized nor started.	Switch fieldbus gateway on and off again. If the error persists, replace the fieldbus gateway or contact SEW Service.
Error while initializing CAN 1 or CAN 2 system bus.	239.10710	Process data are not initialized.	Check the SBus installation and the terminating resistors. Check whether several slave units use the same SBus addresses.

13.1.3 Error during unit replacement

Error			
Description	**Number (hex)**	**Response**	**Remedy**
Error during data backup: Error while accessing memory.	121.28	Fieldbus gateway initializes normally and starts the process data. Restore function not ensured.	Remove write protection from SD memory card.
Invalid data in memory.	121.29		Repeat the "data backup" function
Error during automatic update: Error while reading UUID (Universally Unique Identifier) of slave unit.	121.[No. of the slave unit]20	Gateway initializes normally and starts the process data. The restore function to this slave unit is not ensured.	Slave does not have UUID: • MOVIDRIVE® B: unit firmware .13 required • MOVITRAC® B: unit firmware .17 required
"Restore" function error. Error while reading data from the SD card for the replaced slave unit.	121.[No. of the slave unit]22		Check whether the new unit has the same SBus address as the unit it replaces.
"Restore function" error: Error while transferring the parameter set to the slave unit.	121.[No. of the slave unit]23		The slave unit must be in "Controller inhibit" condition (with MOVITRAC® B "No enable").
Error during data backup: Error while transferring the parameter set from the slave unit to the SD memory card of the fieldbus gateway.	121.[No. of the slave unit]24		MOVIDRIV E®A and MOVITRAC® 07A do not support this function. Check whether another axis-to-axis communication in addition to the gateway communication takes place using the same CAN bus. Use the second CAN bus of MOVIDRIV E® B or MOVIAXIS® for this axis-to-axis communication.

13.2 Diagnostic procedure for operation on EtherNet/IP and Modbus/TCP

The diagnostic procedures described in the following section demonstrate the integration of the UFR41B option into an Ethernet network and the error analysis method for the following problems:

- The UFR41B fieldbus gateway is not integrated properly in the EtherNet/IP or Modbus/TCP network
- The master (scanner) cannot specify any process data.

For more diagnostic information, refer to the online status display in the EtherNet/IP master (scanner), in the Modbus/TCP master and the corresponding online help.

Step 1: Check the status LEDs of UFR41B and the DIP switch settings

The possible DIP switch settings are described in chapter "Setting the DIP switches". For detailed information on the individual LED states, refer to chapter "Status LEDs of the UFR41B option". The following table lists the resulting unit statuses for communication via X30-1 and X30-2 and possible causes. An "X" indicates that the state of the respective LED is not relevant.

LED			
L14 MODULE STATUS	**L13 NETWORK STATUS**	**Operating state**	**Cause**
Off	Off	Off	No voltage supply.
Red	Red	Reset	UFR41B is in reset status.
Red	X	Error	Internal UFR41B fault.
Flashing green	Off	IP stack starting	If DHCP is active, the UFR41B remains in this state until assigned an IP address.
Flashing red	Red	IP Conflict	Conflict with the IP address; another station in the network has the same IP address
Flashing green/red	Flashing green/red	LED Test	All LED conditions are briefly activated.
Flashing green	Flashing green	Application starting	All functions of the UFR41B (e. g. process data and connections to the master) are now active.
Green	Flashing green	Operational	The UFR41B is active on the fieldbus but without a controlling connection to the master.
Green	Green	Connected	There is a controlling connection to a master.
Green	Flashing red	Timeout	A previously controlling connection is in timeout state.

In order to check and set the IP parameters, you can proceed according to section "Setting IP address parameters" or use MOVITOOLS® MotionStudio.

The PING and IPCONFIG commands that you can enter via the DOS console on your PC are further tools for checking the communication via Ethernet.

Step 2: Check the status LED and the status display on the master (scanner)

To do so, use the documentation of the controller or master module.

Should there be no working EtherNet/IP or Modbus/TCP master yet, you can use an SEW master simulator for testing or starting up the UFR41B option. The latest version of the master simulator is available on the SEW website.

You can use the SEW master simulator to exchange process or parameter data with EtherNet/IP or Modbus/TCP profile with an SEW fieldbus interface.

Step 3: Error diagnostics

If UFR41B is in "connected" status, data exchange between master (scanner) and slave is active. If the data is not transferred correctly to the UFR41B fieldbus gateway or lower-level drives via EtherNet/IP or Modbus/TCP, the following steps will assist you in finding the cause for the problem.

A Are the correct values for the process data words displayed in the Gateway Configurator?

If yes, continue with F.

B Is the process data exchange activated in the scanner (master)?

C Is the process data written to the correct location in the scanner? Check the tags and scanner mapping.

D Is the PLC in RUN mode or does active forcing overwrite the transfer of the normal process data words?

E If the PLC does not transmit data to UFR41B, refer to the documentation of the PLC manufacturer for support.

F Was the UFR41B fieldbus gateway configured properly and are all drives configured in the UFR41B online?

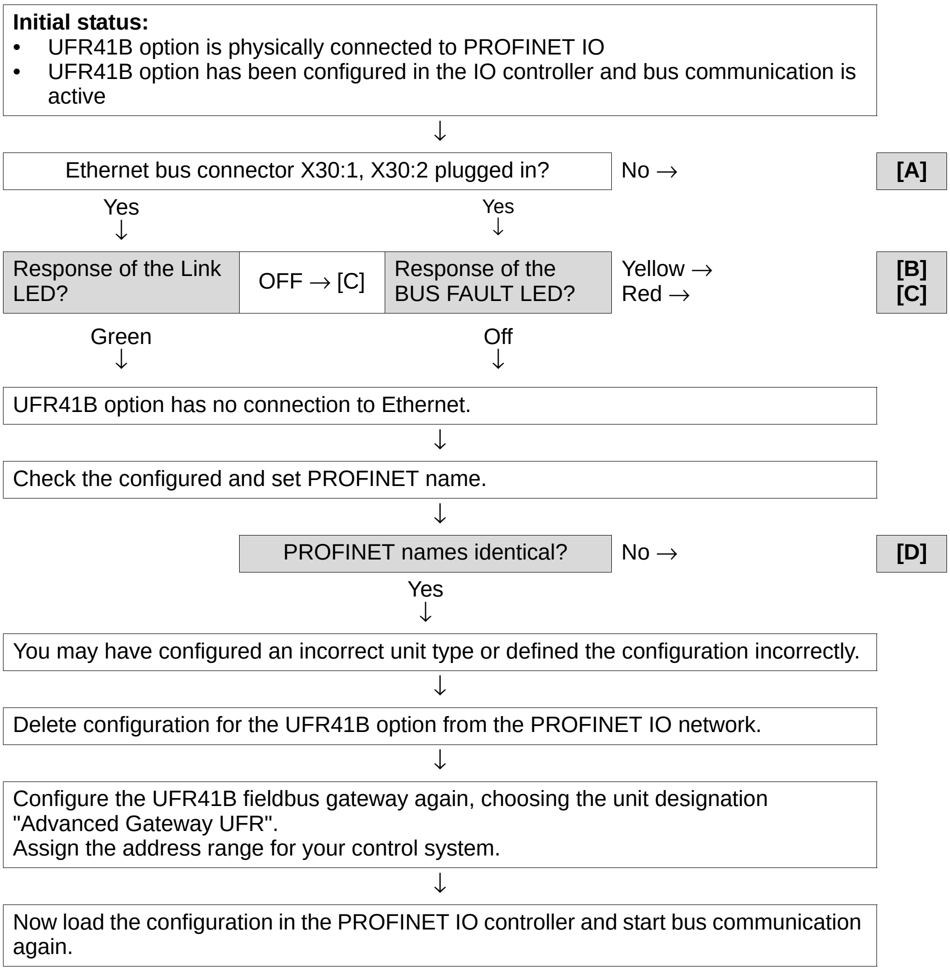
G What errors are indicated in the status displays of the Gateway Configurator?

13.3 Diagnostic procedure for operation on PROFINET IO

The diagnostic procedures described in the following section demonstrate the error analysis methods for the following problems:

- The UFR41B fieldbus gateway does not operate on PROFINET IO.

13.3.1 Diagnostic problem: The UFR41B fieldbus gateway does not operate on PROFINET IO



Initial status:

- UFR41B option is physically connected to PROFINET IO
- UFR41B option has been configured in the IO controller and bus communication is active

↓

Ethernet bus connector X30:1, X30:2 plugged in? No → **[A]**

Yes ↓ Yes ↓

Response of the Link LED? OFF → [C] Response of the BUS FAULT LED? Yellow → Red → **[B]** **[C]**

Green ↓ Off ↓

UFR41B option has no connection to Ethernet.

↓

Check the configured and set PROFINET name.

↓

PROFINET names identical? No → **[D]**

Yes ↓

You may have configured an incorrect unit type or defined the configuration incorrectly.

↓

Delete configuration for the UFR41B option from the PROFINET IO network.

↓

Configure the UFR41B fieldbus gateway again, choosing the unit designation "Advanced Gateway UFR".
Assign the address range for your control system.

↓

Now load the configuration in the PROFINET IO controller and start bus communication again.

[A]	Check the bus cabling.
[B]	The UFR41B option was not yet configured or configured incorrectly. Check the configuration, particularly the device name and the IP address.
	↓
[C]	The UFR41B option indicates that the PROFINET IO controller has not yet established communication.
	↓
	The PROFINET IO controller is switched off or has not yet been started up.
[D]	Adapt PROFINET names

14 Technical Data

14.1 General technical data

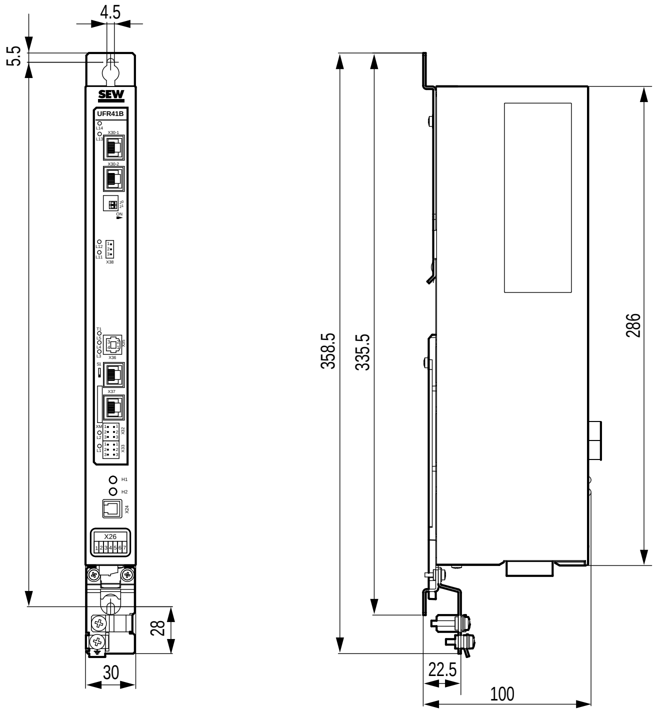
Part number	1821 626 9 (UFR41B without gateway housing UOH21B)
Interference immunity	Meets EN 61800-3
Ambient temperature	Installed in the MOVIAXIS® master module: • 0 °C - +45 °C In the UOH21B gateway housing: • (-10 °C) - +60 °C
Climate class	EN 60721-3-3, class 3K3
Storage temperature	(-25 °C) - +70 °C
Climate class	EN 60721-3-3, class 3K3
Type of cooling	Convection cooling
Degree of protection	IP20
Pollution class	2 according to IEC 60664-1 (VDE0110-1)
Installation altitude	max. 4000 m (NN)

14.2 UFR41B fieldbus gateway

UFR41B fieldbus gateway	
Electrical supply	Integrated in MOVIAXIS® master module (MXM) or in UOH21B gateway housing: • Power consumption: P_{max} = 12 W • U = DC 24 V (-15 % / +20 %) • I_{max} = 900 mA • The UFR41B fieldbus gateway can be supplied from the MOVIAXIS® switched-mode power supply (MXS) or from an external voltage source. To do so, connect X5 between the individual units. • If the UFR41B fieldbus gateway is supplied with DC 24 V from the MOVIAXIS® switched-mode power supply, then the function of the UFR41B fieldbus gateway is ensured when power supply is switched off (external DC 24 V supply at X16 of the MOVIAXIS® switched-mode power supply).
Potential levels	The UFR41B fieldbus gateway has the following potential levels: • Potential control / CAN 1 • Potential Ethernet X30-1 • Potential Ethernet X30-2 • Potential system bus CAN 2
Memory	• Program memory: 8 MByte • Data memory: 4 MByte • Retain data: 32 kByte • System variables (retain): 8 kByte
System bus CAN 2 X32:1 - X32:3 **System bus CAN 1** X33:1 - X33:3	• System bus CAN 1 and CAN 2 to CAN specification 2.0, parts A and B, transmission technology to ISO 11898 • The CAN 2 system bus is electrically isolated • Max. 64 stations per CAN system bus • Address range 0 - 63 • Baud rate: 125 kBaud - 1 MBaud • If X32 or X33 is the bus terminator, you must connect a terminating resistor (120 Ω) externally. • You can remove connector X32 or X33 without interrupting the system bus.
Ethernet 1	System bus, system bus SBUS^{plus} (in preparation)
Ethernet 2	• TCP/IP • Connection options: Engineering PC, other controller, Intranet
USB	USB 1.0 for connecting an engineering PC
SD memory card OMG4.B	• PC-readable • Includes: – Firmware – Gateway application – Data • At least 128 MB memory
Engineering	Engineering takes place using the Ethernet interface (X30-1, X302 and X37) or USB (X35) The engineering of all SEW components connected to the UFR41B fieldbus gateway can be carried out using the UFR41B option. • Engineering software MOVITOOLS® MotionStudio V5.5x

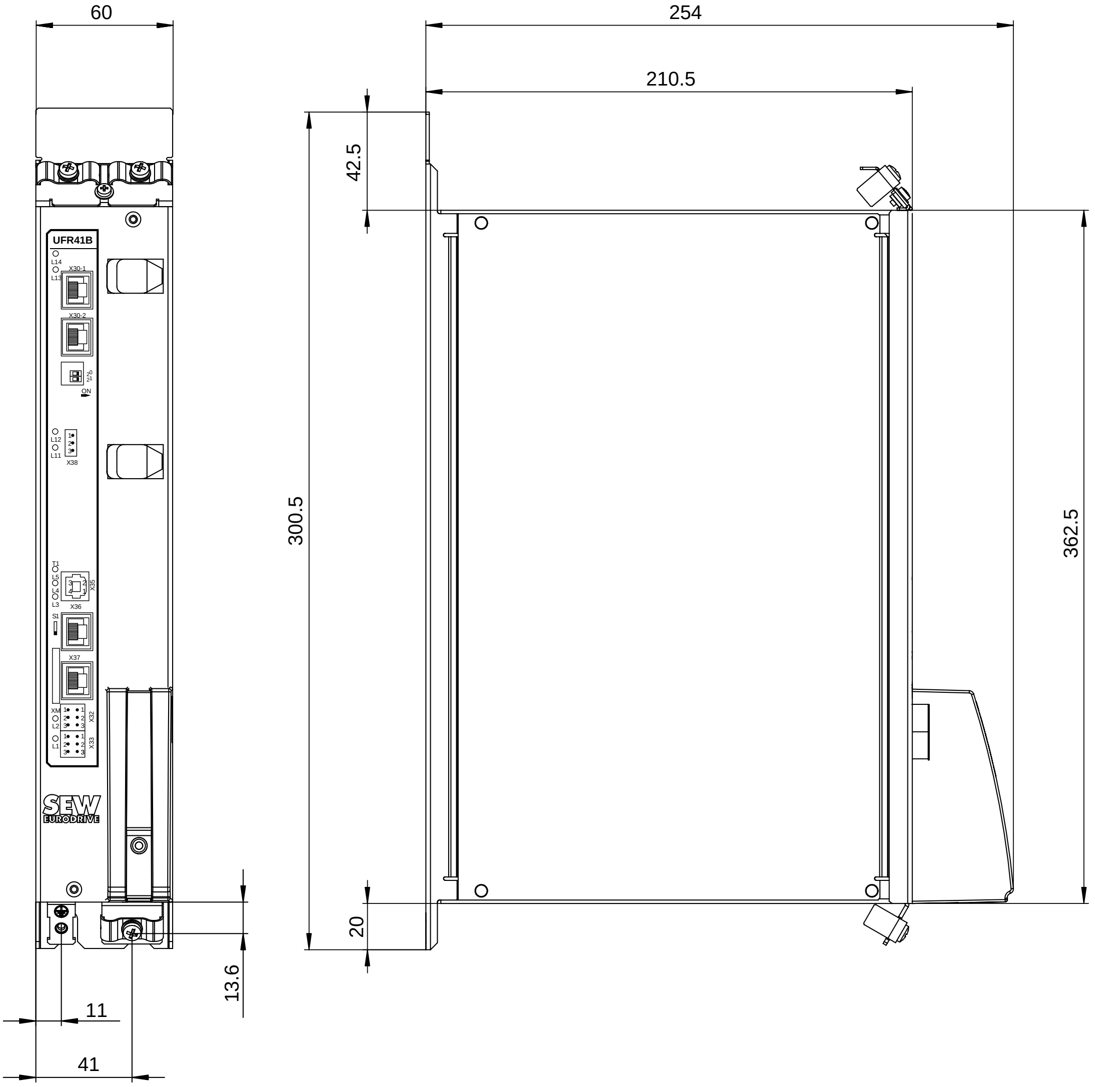
14.3 Dimension drawings

14.3.1 Dimension drawings for fieldbus gateway UFR41B / UOH21B



65341AXX

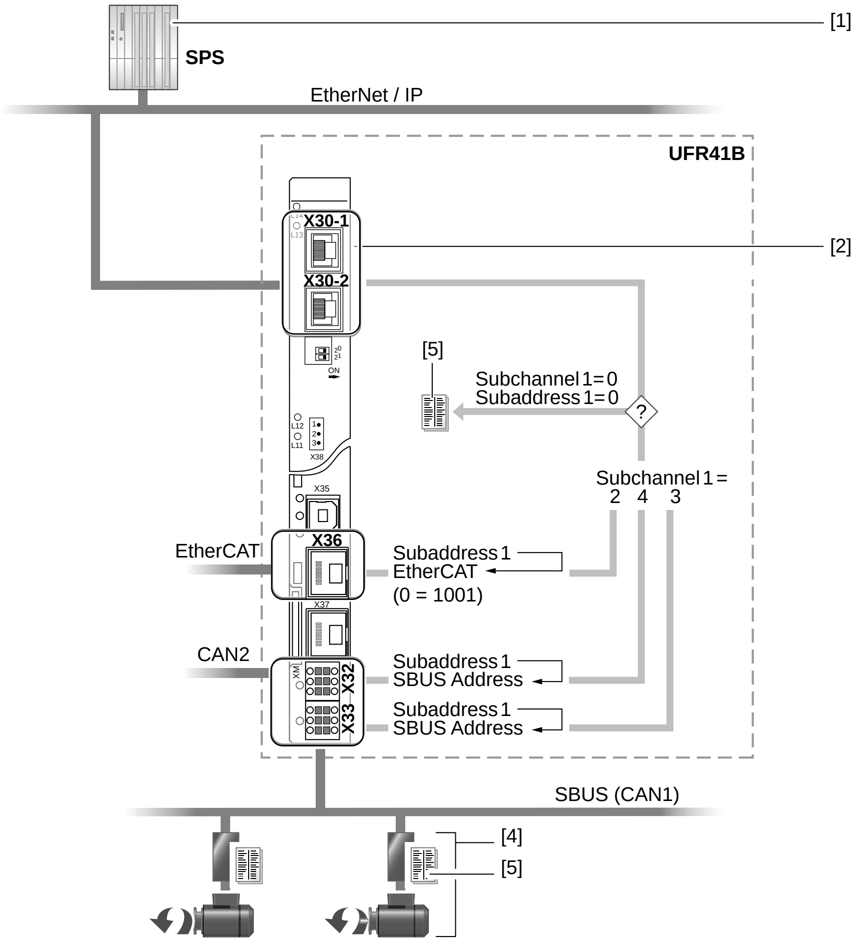
14.3.2 Dimension drawing for MOVIAXIS® master module MXM / UFR41B



65346AXX

15 Appendix

15.1 Parameter access to lower-level units via EtherNet/IP



65338AXX

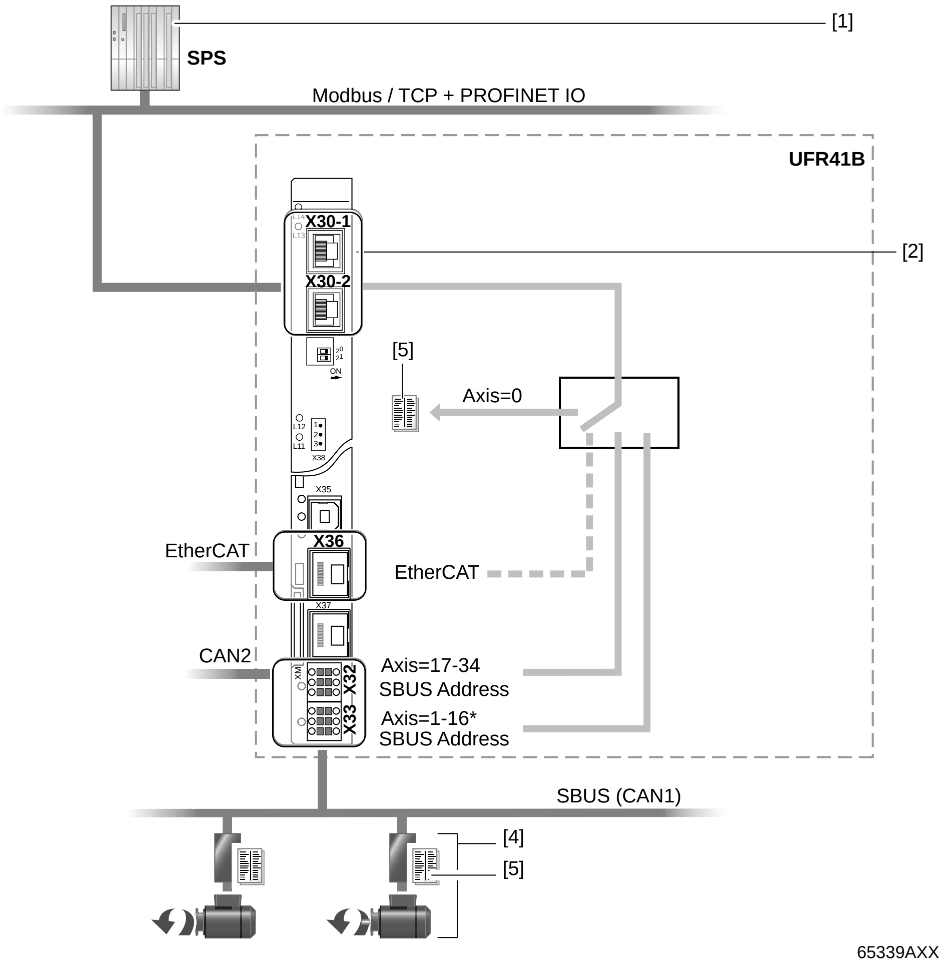
[1] PLC with EtherNet/IP scanner (master)

[2] Industrial Ethernet interface

[4] SEW inverter with SBus interface

[5] Index and parameter list of the unit

15.2 Parameter access to lower-level units via Modbus/TCP or PROFINET



* Slave SBus address 15 must not be used.

"Axis" with PROFINET corresponds to "unit ID" with Modbus/TCP

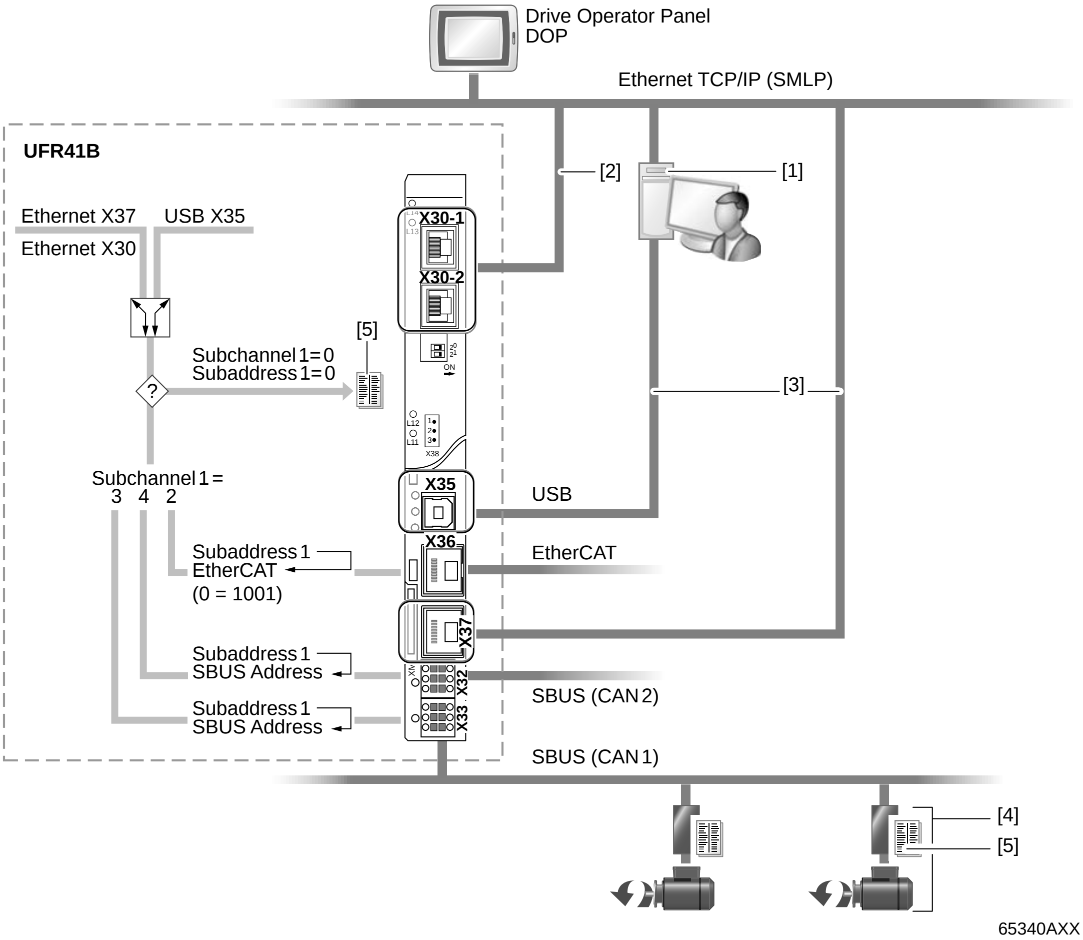
[1] PLC with Modbus scanner

[2] Industrial Ethernet interface

[4] SEW inverter with SBus interface

[5] Index and parameter list of the unit

15.3 Parameter access to lower-level units via engineering interface



[1] Engineering PC

[2] Industrial Ethernet interface (for engineering)

[3] USB/Ethernet engineering interface

[4] SEW inverter with SBus interface

[5] Index and parameter list of the unit