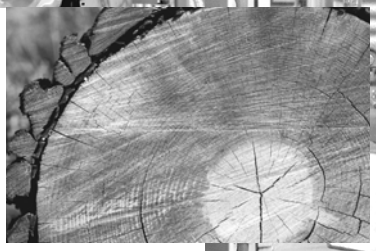
SEW EURODRIVE

MOVI-PLC® *advanced* DHF41B Controller DeviceNet and PROFIBUS DP-V1 Fieldbus Interfaces

Edition 03/2009

16658426 / EN

Manual



SEW EURODRIVE

1 General Information

1.1 How to use the documentation

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

1.2 Structure of the safety notes

The safety notes in this documentation are structured as follows:

Pictogram	SIGNAL WORD
	Type and source of danger. Possible consequence(s) if 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
	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 documentation. Read the documentation before you start working with the unit!

Make sure that the documentation is available to persons responsible for the system and its operation as well as to persons who work independently on the unit. You must also ensure that the documentation is legible.

1.4 Exclusion of liability

You must observe this documentation and the documentation of the connected units from SEW-EURODRIVE to ensure safe operation and to achieve the specified product characteristics and performance requirements. 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

2 Safety Notes

2.1 Other applicable documentation

Note also the following documentation:

- "MOVI-PLC ® *advanced* DHE41B/DHF41B/DHR41B Controller" manual
- "MOVI-PLC® Programming in the PLC Editor" manual

The following publications and documents apply to the connected units:

- Operating instructions of the units
  (Units are, for example, MOVIDRIVE® B, MOVITRAC® B, MOVIAXIS®)
- For units with functional safety technology, also the respective
  "Safe Disconnection - Conditions" manuals

2.2 Bus systems

MOVI-PLC® *advanced* DHF41B supports various bus systems. A bus system makes it is possible to adapt frequency inverters to the particulars of the machinery within wide limits. 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, though not uncontrolled, system behavior.

2.3 Safety functions

The MOVIDRIVE® MDX60B/61B and MOVITRAC® B inverters may not perform safety functions without higher-level 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® MDX60B/61B, MOVITRAC® B".

2.4 Hoist applications

MOVIDRIVE® MDX60B/61B and MOVITRAC® B are not designed for use 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:

- Startup MOVI-PLC® *advanced* DHF41B on the DeviceNet fieldbus system and PROFIBUS DP-V1.
- Configure the DeviceNet master with EDS files.
- Configure the PROFIBUS DP-V1 master using GSD files.

The creation of IEC programs or the connection of SEW drives to the system bus interfaces of MOVI-PLC® is not described.

3.2 Characteristics

The powerful, universal fieldbus interfaces of the DHF41B option enable you to use the option to connect to higher-level automation systems via DeviceNet and PROFIBUS DP-V1.

3.2.1 Process data exchange

MOVI-PLC- *advanced* DHF41B offers digital access to a special data range via the DeviceNet and PROFIBUS interfaces. This data range is evaluated by IEC 61131-3 as process input and output data to a higher-level controller. The meaning of the transferred data depends on the IEC program.

3.2.2 Parameter access

This parameter data exchange enables you to implement applications for which all important parameters are stored in the higher-level programmable controller, so that there is no need to set parameters manually in the MOVI-PLC® *advanced* DHF41B.

In DeviceNet operation, the parameters of the inverter are set solely via *explicit messages*.

In PROFIBUS operation, two parameter access options are available:

- The 8 byte MOVILINK® parameter channel enables read and write functions to the most important drive parameters.
- The PROFIBUS DP-V1 parameter mechanism offers universal access to any unit information.

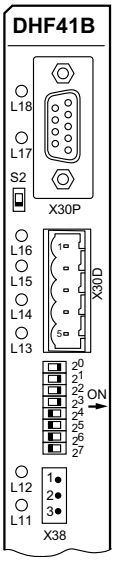
3.2.3 Monitoring functions

Using a fieldbus system requires additional monitoring functions, for example, time monitoring of the fieldbus (fieldbus timeout) or rapid stop concepts. For example, you can adapt the monitoring functions specifically to your application in the IEC program. You can determine, for instance, which fault responses should be triggered in the event of a bus error. For many applications, a rapid stop function is useful. However, you can also freeze the last setpoints so that the drive continues to operate with the most recently valid setpoints. As the range of functions for the control terminals is also available in fieldbus mode, you can continue to implement rapid stop concepts using the terminals of MOVI-PLC® *advanced* DHF41B, irrespective of the fieldbus used.

4 Assembly and Installation Notes on the DeviceNet Fieldbus

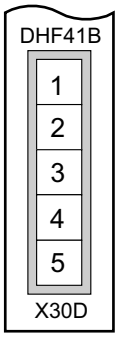
4.1 Connecting MOVI-PLC® advanced DHF41B to a DeviceNet network

The following sections describes the terminals, DIP switches, and LEDs relevant for DeviceNet fieldbus operation.

Front view MOVI-PLC® *advanced* DHF41B	Designation	LED DIP switch Terminal		Function
 63598AXX	**LED**	**LED 18** **LED 17** **LED 16** **LED 15** **LED 14** **LED 13** **LED 12** **LED 11**	 Mod/Net Polled I/O Bit strobe I/O BUS FAULT - -	LEDs 17 and 18 are reserved for PROFIBUS. LED 18 is lit orange: DHF41B option is being initialized The two-color LEDs 13 ... LED 16 indicate the current status of the fieldbus interface and the DeviceNet system (see chapter "Status LED of the DHF41B option"). Reserved Reserved
	Connector X30D: **DeviceNet** **(plug-in terminals)**	**X30D:1** **X30D:2** **X30D:3** **X30D:4** **X30D:5**	V- CAN_L DRAIN CAN_H V+	0V24 CAN_L DRAIN CAN_H 24 V
	DIP switch S2 **Switching between PROFIBUS and DeviceNet**	**S2**	Top Bottom	Fieldbus interface PROFIBUS (X30P) active Fieldbus interface DeviceNet (X30P) active
	For operation via DeviceNet: **DIP switch** **for setting** **MAC-ID and baud rate**	**2^0** **2^1** **2^2** **2^3** **2^4** **2^5** **2^6** **2^7**		The DIP switches 2^0 - 2^5 are used to set the MAC-ID (**M**edia **A**ccess **C**ontrol **Id**entifier). The MAC-ID represents the node address (address range 0 - 63) Setting the baud rate Setting the baud rate
	Connector X38: **CAN for safety-relevant communication** **(plug-in terminals)**	**X38:1** **X38:2** **X38:3**		Reserved Reserved Reserved

4.2 Connecting DeviceNet (connector X30D)

The assignment of connecting terminals is described in the DeviceNet specification (Volume I, Appendix A).



61612AXX

The DHF41B option is opto-decoupled on the driver side in accordance with the DeviceNet specification (Volume I, Chapter 9). This means the CAN bus driver must be powered with 24 V voltage via the bus cable. The cable to be used is also described in the DeviceNet specification (Volume I, Appendix B).

Make the connection according to the color code specified in the following table.

Pin no.	Signal	Meaning	Color coding
1	V-	0V24	BK
2	CAN_L	CAN_L	BU
3	DRAIN	DRAIN	blank
4	CAN_H	CAN_H	WH
5	V+	24 V	RD

Connecting DHF41B to the DeviceNet

According to the DeviceNet specification, a linear bus structure without or with very short droplines is required.

The maximum permitted cable length depends on the baud rate setting:

Baud rate	Maximum cable length
500 kBd	100 m
250 kBaud	250 m
125 kBd	500 m

4.3 Shielding and routing bus cables

The DeviceNet interface supports RS-485 communications protocol and requires cable type A specified for DeviceNet in accordance with EN 50170 as shielded, twisted-pair cable for the physical connection.

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.
- Apply the shielding of the bus cable on both ends over a large area.
- 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

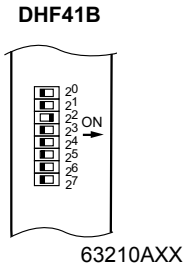
In case of fluctuations in the ground potential, a compensating current may flow via the bilaterally connected shield that is also connected to the protective earth (PE). Make sure you supply adequate equipotential bonding in accordance with relevant VDE regulations in such a case.

4.4 Bus termination

In order to avoid disruptions in the bus system due to reflections, each DeviceNet segment must be terminated with 120-ohm bus terminating resistors at the first and last physical station. Connect the bus terminating resistor between connections 2 and 4 of the bus plug.

4.5 Setting the DIP switches

	TIP
	Before each change to the DIP switches, disconnect MOVI-PLC® *advanced* DHF41B from the voltage supply. The DIP switch settings are adopted during initialization only.



2^0 - 2^5 = MAC ID setting
2^6 - 2^7 = Baud rate setting

Setting the MAC-ID

The MAC ID (**M**edia **A**ccess **C**ontrol **Id**entifier) is set on the DHF41B option using DIP switches 2^0 - 2^5 in a binary coded manner. The MAC-ID represents the node address of the DHF41B. The DHF41B supports the address range 0 - 63.

Setting the baud rate

The baud rate is set using DIP switches 2^6 and 2^7.

DIP switch		Baud rate
2^6	2^7	
0	0	125 kBd
1	0	250 kBd
0	1	500 kBd
1	1	Invalid

A maximum of 64 DeviceNet data words can be exchanged between the DeviceNet module and the DHF41B option. The number is set from the IEC program either using the MC_DeviceNetPDConfig function block or the DeviceNet scanner.

4.6 Status LED of the DHF41B option

The DHF41B option card is equipped with four two-color LEDs for diagnosing the DeviceNet system. They indicate the current status of the DHF41B and the DeviceNet system. The unit status corresponding to the status of the LED is described in chapter "Error diagnostics".

LED		
Designation	**Abbreviation**	**Complete LED designation**
L16	MOD/NET	Module/Network status
L15	PIO	Polled I/O
L14	BIO	Bit-strobe IO
L13	BUS FAULT	BUS FAULT

LED L16 (Mod/Net)

The function of the **L16 LED (Mod/Net** = Module/Network Status) described in the following table is specified in the DeviceNet specification.

Status of the L16 LED	Status	Meaning
Off	Not switched on / offline	• Unit is offline • Unit performs DUP MAC check • Unit is switched off
Flashing green (1 s cycle)	Online,in operational mode	• The unit is online and no connection has been established • DUP-MAC check performed successfully • A connection has not yet been established with a master • Missing, incorrect or incomplete configuration
Lights up green	Online, operational mode and connected	• Online • Connection to a master has been established • Connection is active (established state)
Flashing red (1 s cycle)	Minor fault or connection timeout	• A correctable fault has occurred • Polled I/O and/or bit strobe I/O connections are in timeout status • DUP-MAC check has detected an error
Red light	Critical fault or critical link failure	• A correctable fault has occurred • BusOff • DUP-MAC check has detected an error

L15 LED (PIO)

The **L15 (Polled I/O)** LED monitors the polled I/O connection.

Status of the L15 LED	Status	Meaning
Flashing green (125 ms cycle)	DUP-MAC check	Unit is performing DUP-MAC check
Off	Not switched on / offline but not DUP-MAC check	• Unit is offline • Unit is switched off
Flashing green (1 s cycle)	Online, in operational mode	• Unit is online • DUP-MAC check performed successfully • A polled IO connection is being established with a master (configuring state) • Missing, incorrect or incomplete configuration
Lights up green	Online, operational mode and connected	• Online • A polled I/O connection has been established (established state)
Flashing red (1 s cycle)	Minor fault or connection timeout	• Invalid baud rate setting via DIP switches • A correctable fault has occurred • Polled I/O connection is in timeout status
Red light	Critical fault or critical link failure	• A fault that cannot be remedied has occurred • BusOff • DUP-MAC check has detected an error

L14 LED (BIO)

The **L14 (Bit-strobe I/O)** LED monitors the bit-strobe I/O connection.

Status of the L14 LED	Status	Meaning
Flashing green (125 ms cycle)	DUP-MAC check	Unit is performing DUP-MAC check
Off	Not switched on / offline but not DUP-MAC check	• Unit is offline • Unit is switched off
Flashing green (1 s cycle)	Online and in operational mode	• Unit is online • DUP-MAC check performed successfully • A BIO connection is being established with a master (configuring state) • Missing, incorrect or incomplete configuration
Lights up green	Online, operational mode and connected	• Online • A BIO connection has been established (established state)
Flashing red (1 s cycle)	Minor fault or connection timeout	• Invalid number of process data is set via DIP switches • A correctable fault has occurred • Bit-strobe I/O connection is in timeout state
Red light	Critical fault or critical link failure	• A fault that cannot be remedied has occurred • BusOff • DUP-MAC check has detected an error

L13 LED (BUS FAULT)

The **L13 (BUS-OFF)** LED indicates the physical status of the bus node.

Status of the L13 LED	Status	Meaning
Off	NO ERROR	The number of bus errors is in the normal range (error active status).
Flashing red (125 ms cycle)	BUS WARNING	The unit is performing a DUP-MAC check and cannot send any messages because no other stations are connected to the bus (error passive state)
Flashing red (1 s cycle)		The number of physical bus errors is too high. No more error telegrams are actively written to the bus (error passive state).
Red light	BUS ERROR	• BusOff state • The number of physical bus errors has increased despite a switch to the error-passive state. Access to the bus is switched off.
Yellow light	POWER OFF	External voltage supply has been turned off or is not connected.

Power-UP test

A power-up test of all LEDs is performed once the drive inverter has been switched on. The LEDs are switched on in the following sequence:

Time [ms]	LED L16 MOD/NET	LED L15 PIO	LED L14 BIO	LED L13 BUS FAULT
0	Green	Off	Off	Off
250	Red	Off	Off	Off
500	Off	Green	Off	Off
750	Off	Red	Off	Off
1000	Off	Off	Green	Off
1250	Off	Off	Red	Off
1500	Off	Off	Off	Green
1750	Off	Off	Off	Red
2000	Off	Off	Off	Off

5 Project Planning and Startup on the DeviceNet Fieldbus

This section provides you with information on configuration of the DeviceNet master and startup of MOVI-PLC® *advanced* DHF41B for fieldbus operation.

	TIP
	The current version of the EDS file for the DHF41B option is available on the SEW homepage (http://www.sew-eurodrive.com) under the heading "Software".

5.1 Validity of EDS files for the DHF41B option

	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!

The current EDS file SEW_MOVIPLC_DHF41B.eds is available for configuring the master (DeviceNet scanner) for DHF41B.

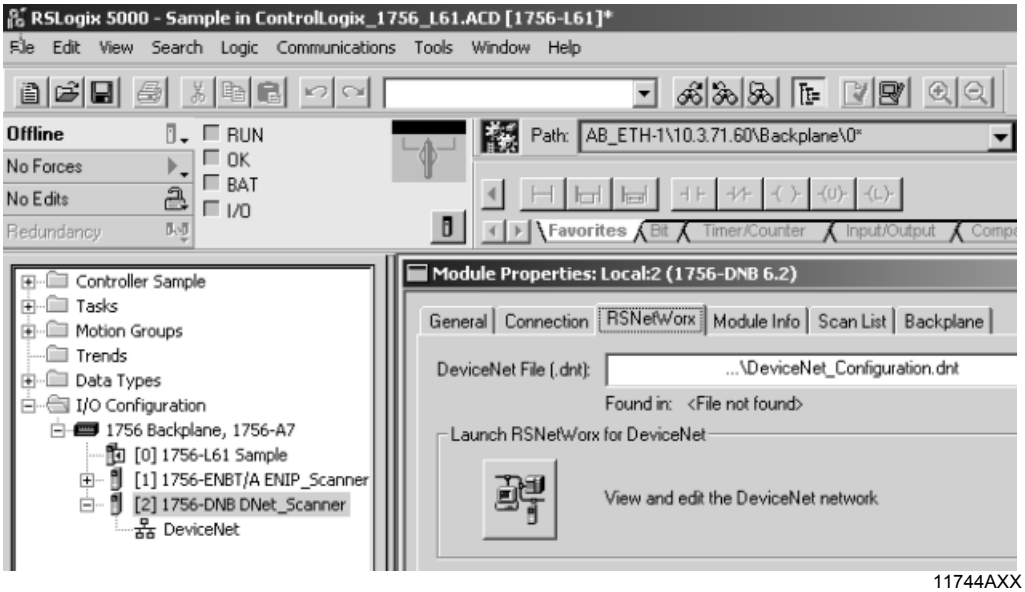
Install the following files using the RSNetWorx software to establish the DeviceNet network with the DHF41B option: Proceed as follows:

- Select <Tools/EDS-Wizard> from the menu in RSNetWorx. You will be prompted to enter the names of the EDS and icon files.
- The files will be installed. For more details on how to install the EDS file, refer to the Allen Bradley documentation for RSNetWorx.
- After installation, the device is available in the device list under the entry "Vendor/SEW EURODRIVE GmbH".

5.2 Configuring PLC and master (DeviceNet scanner)

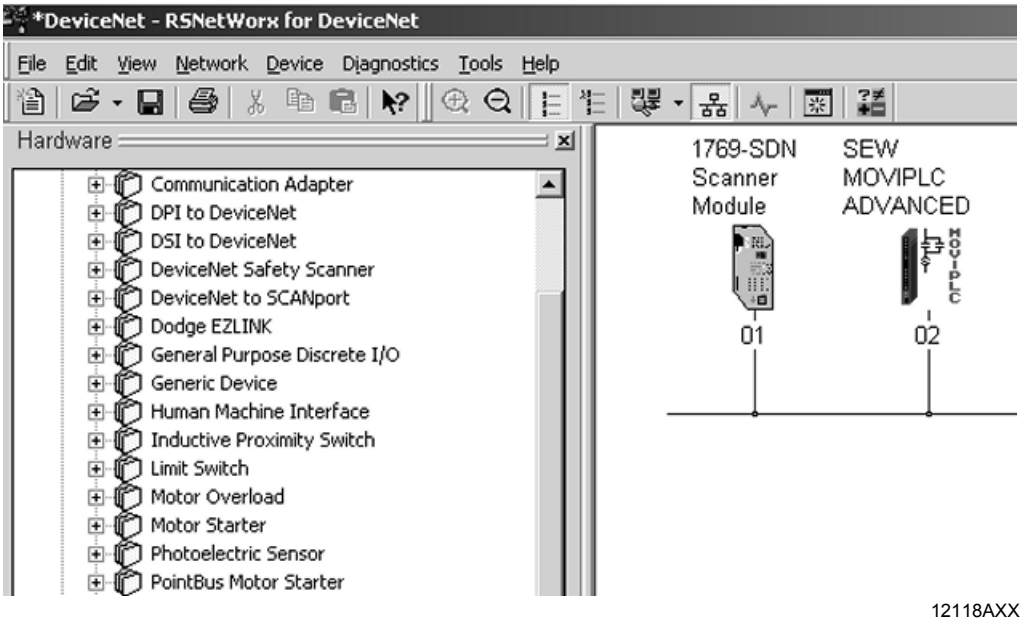
The following samples refer to the usage of an Allen-Bradley-PLC ControlLogix 1756-L61 together with the RSLogix 5000 programming software and the DeviceNet RSNetWorx configuration software for DeviceNet.

After adding the DeviceNet Scanner to the I/O configuration, the file *.dnt containing the DeviceNet configuration is selected. To view and edit the DeviceNet configuration, you can launch RSNetWorx from this dialog (see following figure).



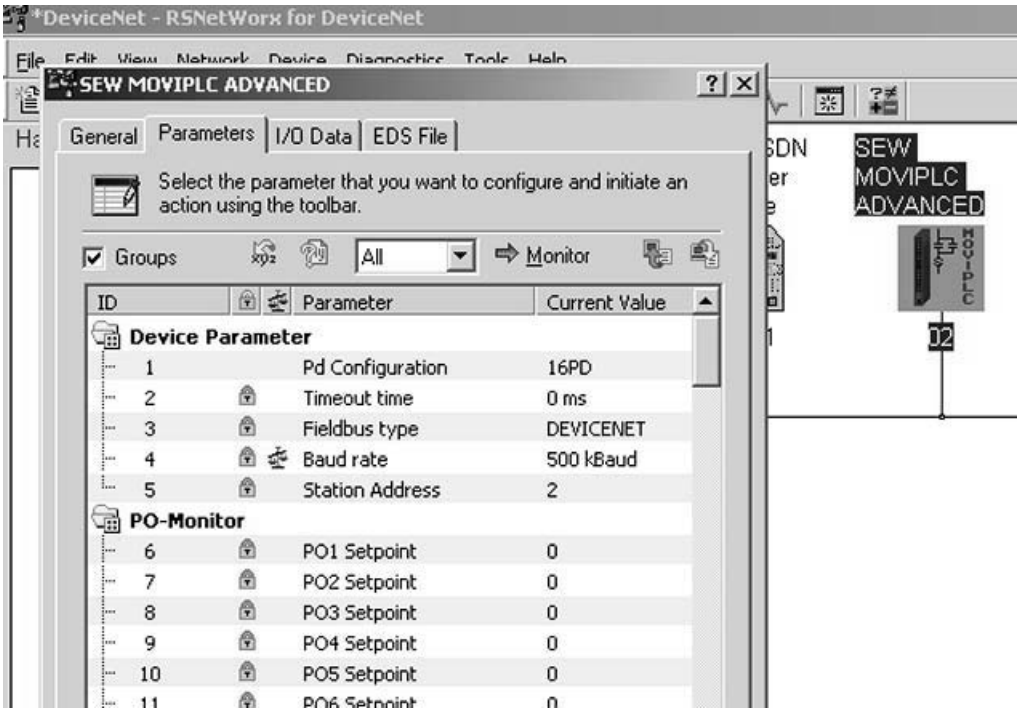
11744AXX

In RSNetWorx for DeviceNet (see following figure), either perform an online scan or add the required devices to the graph by drag and drop. The address given under the icon of the device must be equal to the MAC-ID set by the DIP switches of the DHF41B. If the selection list does not include the required devices, corresponding EDS files have to be registered via [Tools] / [Wizard].



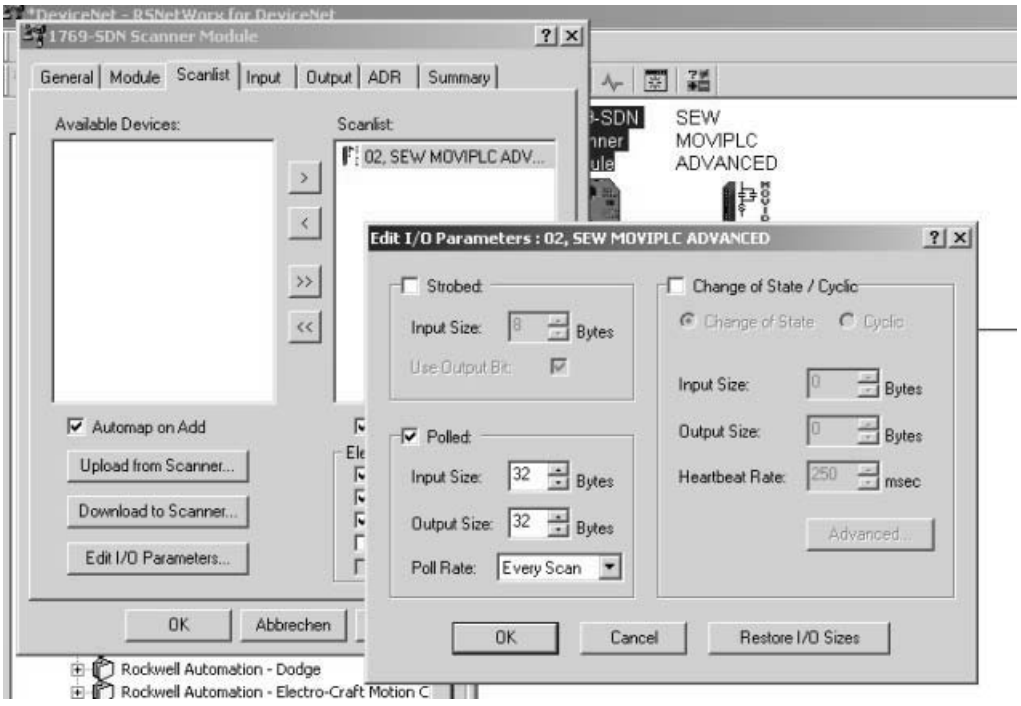
12118AXX

You can check the process data configuration of the DHF41 option by reading the "device properties" in online mode (see following figure).



12119AXX

The "Pd configuration" parameter indicates the number (1 - 64) of process data words (PD) that was set via RSNetWorx or in the IEC program of the MOVI-PLC® using the MC_DeviceNetPDConfig function block (see the chapter "Settings in the MOVI-PLC® *advanced* DHF41B") and determines the I/O parameters for the DeviceNet scanner (see following figure).



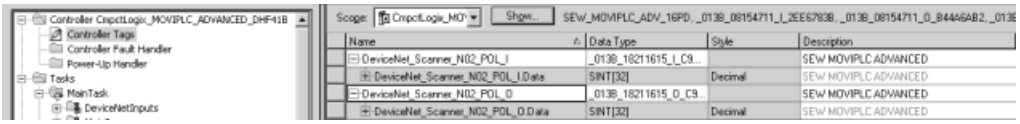
12120AXX

After adding the MOVI-PLC® *advanced* DHF41B to the "scan list", the number of polled I/O bytes must be set to 2 x number of PD (e.g. PD = 16 number of polled input bytes = 32 and output bytes = 32) via "Edit I/O parameters". You can close RSNetWorx once you have saved and downloaded the DeviceNet configuration to the scanner.

Depending on the DeviceNet configuration and the mapping rules in the scanner, the data from and to DeviceNet units is packed into a DINT array that is transferred from the scanner to the local I/O tags of the Logix processor.

In order not to have to search for the data from a certain device in this array manually, the "DeviceNet Tag Generator" tool automatically generates copy commands and two controller tags (input & output as byte arrays) for each DeviceNet device.

The tag name contains the MAC-ID of the DeviceNet unit and *POL_I* for polled input data or *POL_O* for polled output data (see following figure).



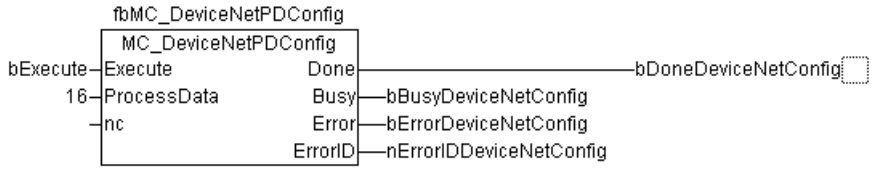
12121AXX

5.3 Settings in MOVI-PLC® advanced DHF41B

For a detailed description on how to create IEC programs, refer to the "MOVI-PLC® *advanced* DHE41B/DHF41B/DHR41B Controller" manual. This section only describes the fieldbus-specific characteristics.

5.3.1 Process data configuration

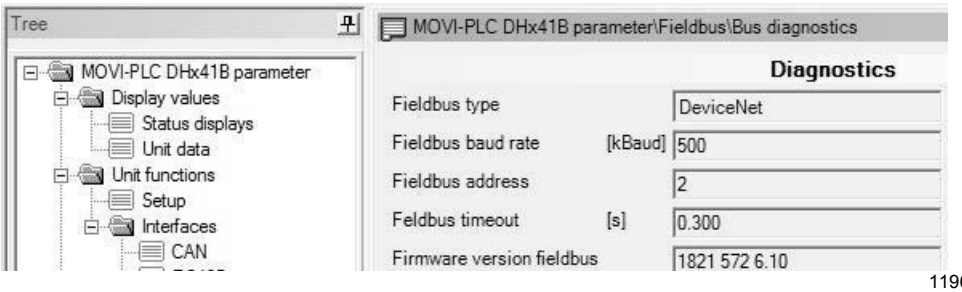
The process data interface is normally configured by the master (scanner). It sets the number of process data words and the timeout interval. The number of process data can also be specified in the IEC program (see following figure) using the *MC_DeviceNetPDConfig* function block.



12045AXX

Setting the *Execute* input to *TRUE* will initialize the fieldbus interface with the value set at the *ProcessData* input.

In the parameter tree of MOVITOOLS® MotionStudio (index 8451), the currently set value is displayed in the field "PD configuration" (see following figure).



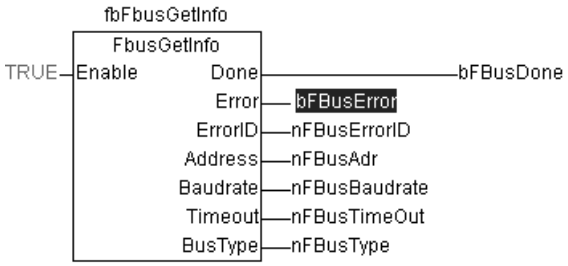
11968AXX

The number of process data words received and sent via DeviceNet is set at the *ProcessData* input. The value entered here must match the number of I/O data, which is configured in the DeviceNet scanner.

Refer to the IEC program and its documentation for information on whether and how the transmitted process data words are used.

Parameterization using the *MC_DeviceNetPDConfig* function block in the IEC program offers the advantage that the process data configuration is stored on the memory card of the MOVI-PLC® *advanced* DHF41B option together with the associated program.

5.3.2 Status of the fieldbus interface



12046AXX

The *FbusGetInfo* function block (or *ProfibusDeviceNetGetInfo*) makes the status and some display parameters of the fieldbus interface available to the IEC program and for diagnostic purposes.

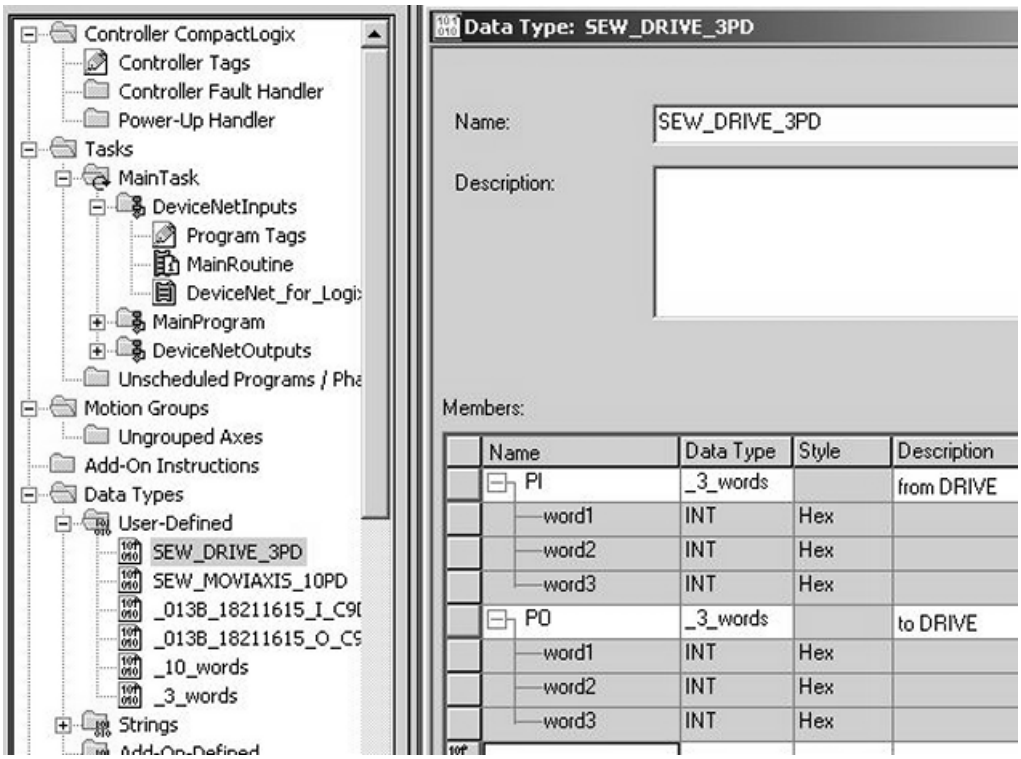
If there is no communication with the fieldbus master, the output *Error* is set to *TRUE*. During an active fieldbus connection, the output *Done* is set to *TRUE*, and the outputs *Address*, *Baud rate*, *Timeout* and *Bus type* show the respective parameters as they were set via the DIP switches of the DHF41B option or via the PLC.

5.4 Project planning examples in RSLogix5000

5.4.1 MOVI-PLC® advanced DHF41B with 16 process data

1. Set the DIP switches on the DHF41B to do the following:
   - Adjust the baud rate to the DeviceNet
   - Set the address (MAC-ID) to a value used by no other node
2. Insert MOVI-PLC® *advanced* DHF41B into the DeviceNet configuration as described in the chapter "Configuring the PLC and master (DeviceNet scanner)".
3. Set the number of process data words of MOVI-PLC® *advanced* DHF41B to 16.
4. 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 inverter's process data (see following figure).

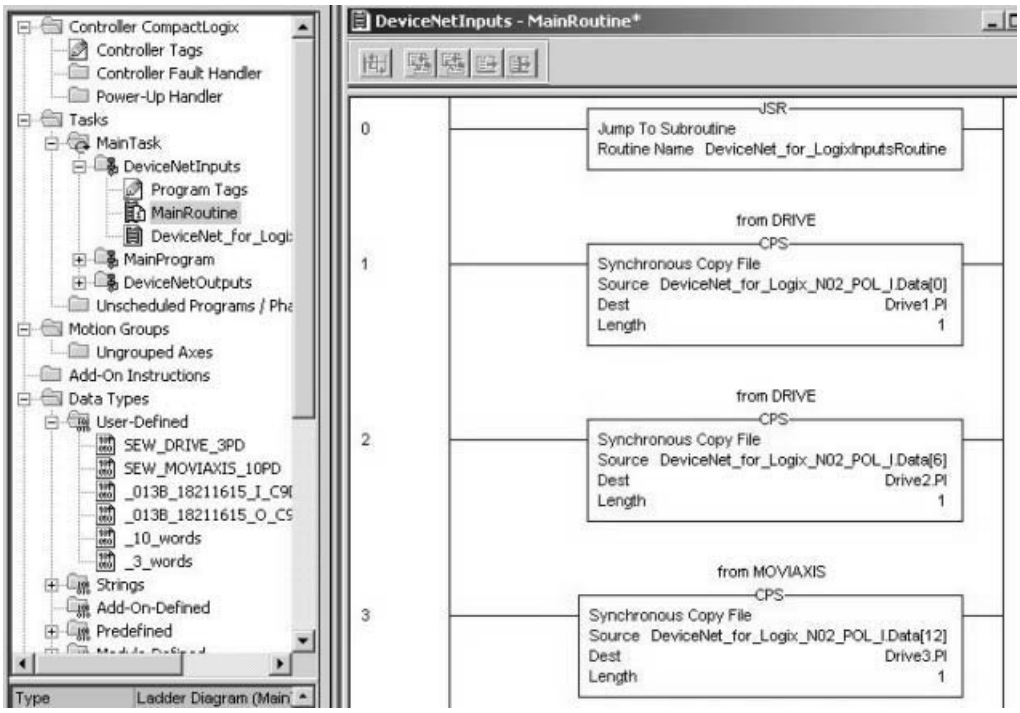


11962AXX

The description for process input and output data of the controller tag can be made in accordance with the definition of the process data (PD) in MOVI-PLC® *advanced* DHF41B.

5. To copy the data of MOVI-PLC® *advanced* DHF41B to the new data structure, a CPS command is added at the "MainRoutine" which reads the data from the local IO (see following figure).

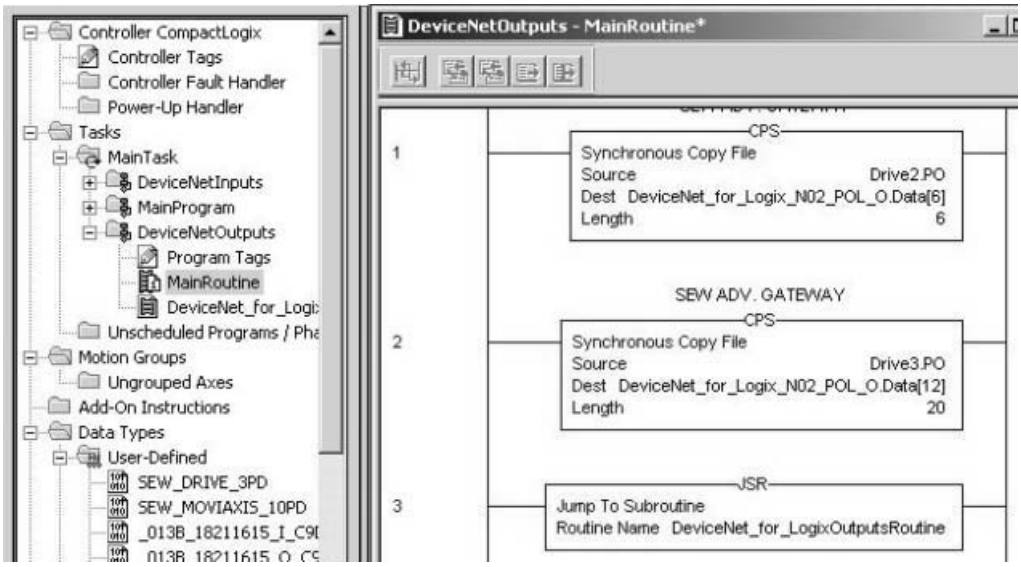
Make sure that this CPS command is executed **after** the automatically (by DeviceNet Tag Generator) generated **DNet_ScannerInputsRoutine**.



11963AXX

In order to copy the data to the MOVI-PLC® *advanced* DHF41B, a CPS command is added to the "MainRoutine" that writes the data to the local I/O.

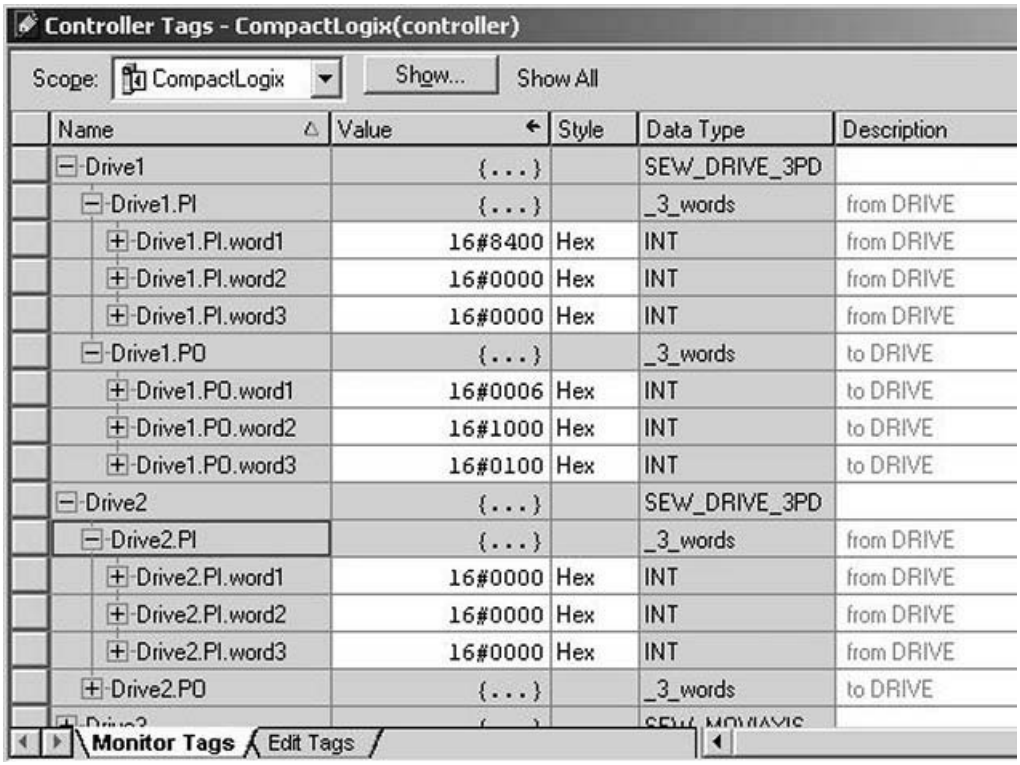
These CPS commands are executed **before** the automatically generated (by DeviceNet Tag Generator) *DNet_Scanner_OutputsRoutine*.



11964AXX

6. Now save the project and transfer it to the PLC. Set the PLC to RUN mode and set the **Scanner CommandRegister.Run** control bit to "1" to activate data exchange via DeviceNet.

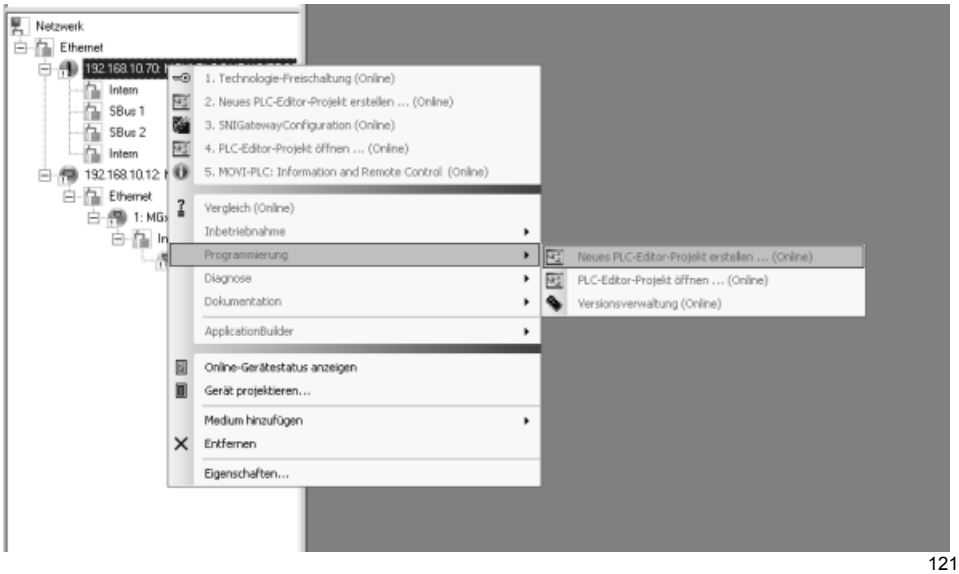
   The MOVI-PLC® *advanced* DHF41B can now read the actual values and write setpoints.



11965AXX

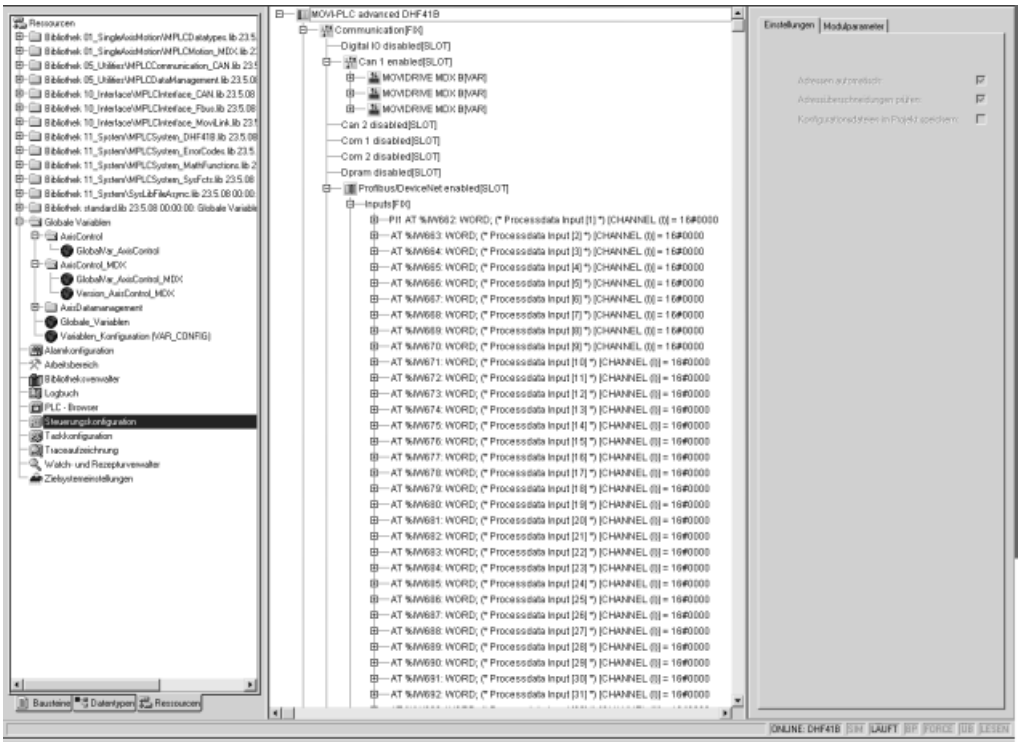
The process data should correspond to the values displayed in the PLC Editor or in the diagnostics plug-in of the active IEC program in MOVITOOLS® MotionStudio. If there is no IEC program in MOVI-PLC®, you can create one as follows:

- Open the context menu of the PLC in MOVITOOLS® MotionStudio and run the project wizard "Create new PLC Editor project" (see following figure).



12115ADE

- Use the wizard to create a new AxisControl project and upload it to MOVI-PLC® *advanced* DHF41B using the menu item "Online login".
- Start the loaded program via the menu item "Online start". You can now monitor the uploaded process data under "Resources Control configuration ". (See following figure).

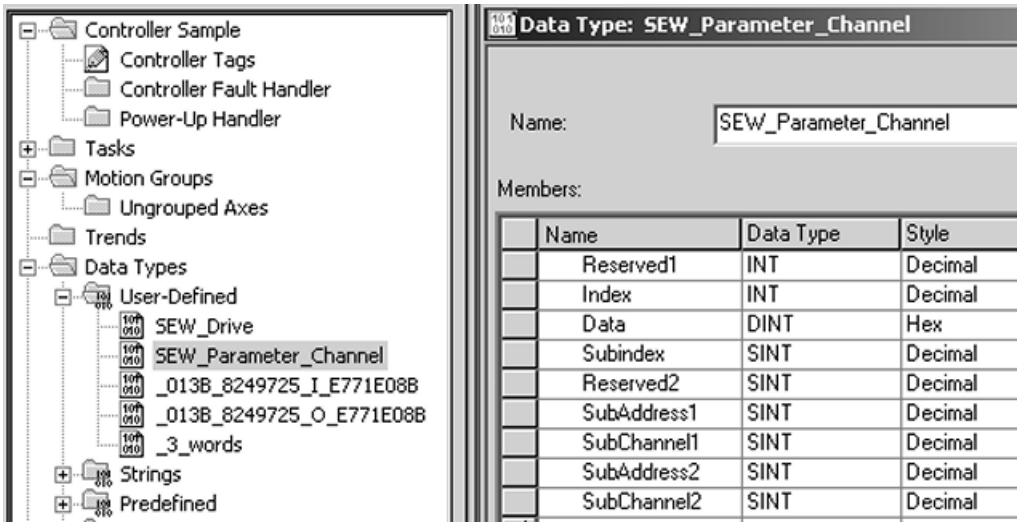


12114ADE

5.4.2 Access to parameters of MOVI-PLC® advanced DHF41B

For easy read access to the parameters of MOVI-PLC® *advanced* DHF41B via *explicit messages* and the *register object*, proceed as follows:

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



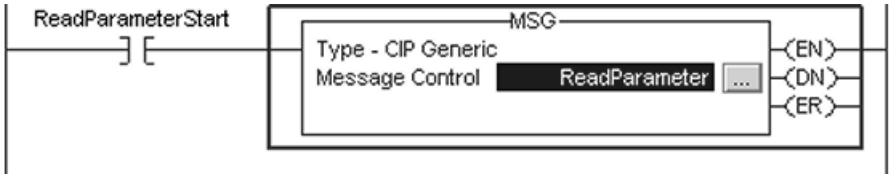
11764AXX

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

Name	Data Type
ReadParameter	MESSAGE
ReadParameterRequest	SEW_Parameter_Channel
ReadParameterResponse	SEW_Parameter_Channel
ReadParameterStart	BOOL

11765AXX

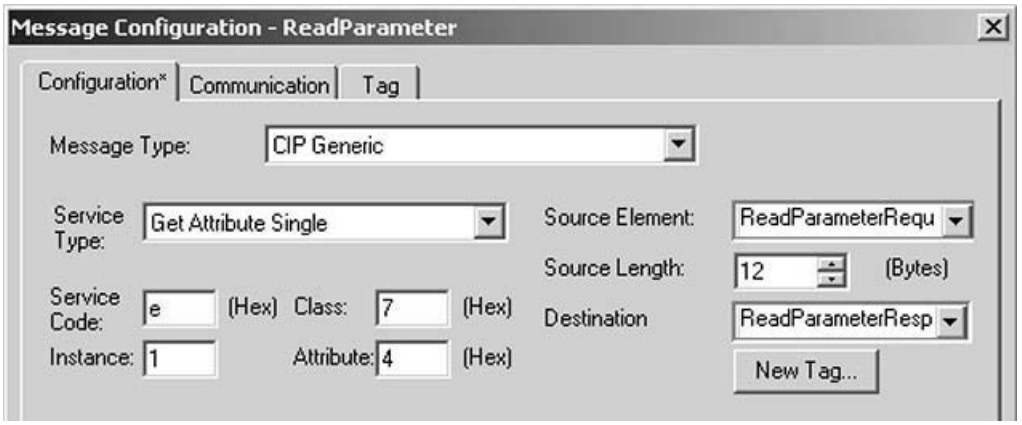
3. Create a rung for the "ReadParameter" execution (following figure).



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).



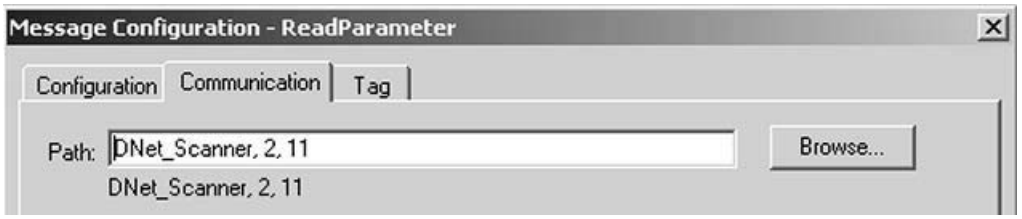
11767AXX

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. The target device is to be specified on the Communication tab (see following figure).



11768AXX

The "path" edit box comprises the following entries:

- Name of the scanner (e. g. DNet_Scanner)
- 2 (always 2)
- Slave address (e. g. 11)

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)

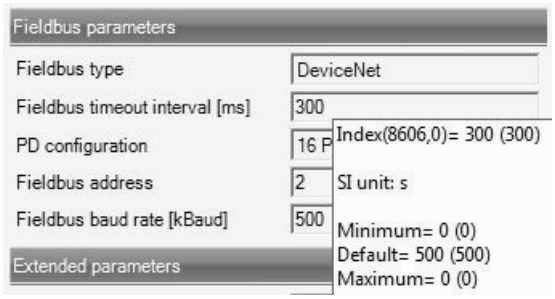
Scope: DeviceNet | Show... | SEW_Parameter_Channel, BOOL, MESSAGE

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 delay of MOVI-PLC® *advanced* DHF41B (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.



11969AXX

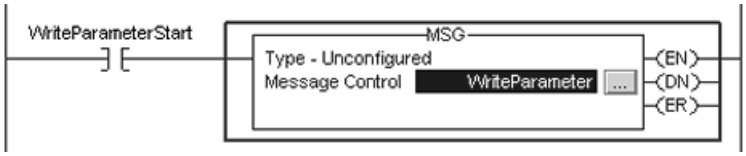
Only a few additions are necessary for activating write access to a parameter:

- Create the controller tags (see following figure)

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

11771AXX

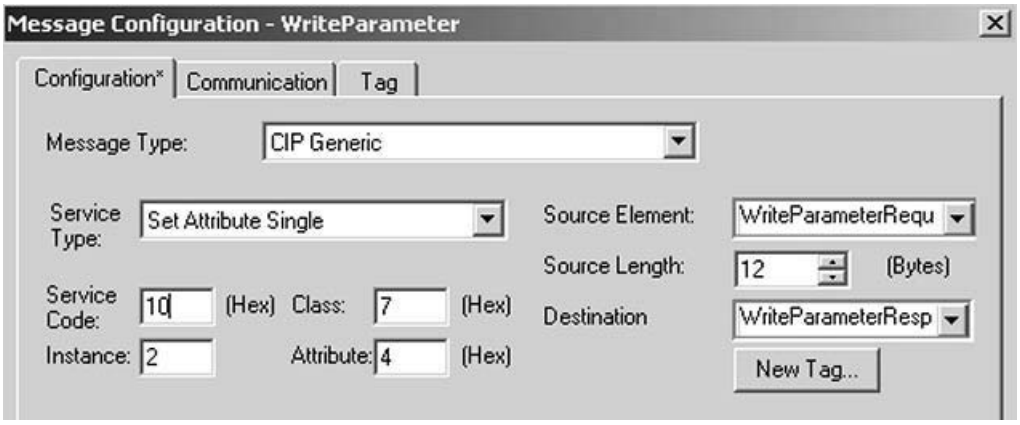
- Create a rung for executing the "WriteParameter" command (see following figure).



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).



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}

7. After downloading the changes to the PLC, index and value to be written into the parameter can be entered at *WriteParameterRequest.Index* and *WriteParameter-Request.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, 22hex (33 dec) was written to index 11001 (H1).

You can check the value in the MOVITOOLS® MotionStudio parameter tree or the PLC Editor. The tooltip of a parameter displays for example index, subindex, factor, etc. of the parameter.

5.4.3 Access to unit parameters of lower-level units

Access to the unit parameters of a MOVITRAC® B, for example, which is connected to the SBus 1 of MOVI-PLC® *advanced* DHF41B is identical with the unit parameter access to MOVI-PLC® *advanced* DHF41B itself (see chapter 5.4.3)

The only difference is that, for example, **Read/WriteParameterRequest.SubChannel1** must be set to **3** and **Read/WriteParameterRequest.SubAddress1** must be set to the SBus address of the MOVITRAC® B unit connected to DHF41B option (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 the DHF41B 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 "Appendix".

6 DeviceNet Operating Characteristics

6.1 Process data exchange

Polled I/O

The polled I/O messages correspond to the process data telegrams sent to the controller MOVI-PLC® *advanced* DHF41B. A maximum of 64 process data words can be exchanged between the controller and the DHF41B option in this case. The process data length is set, for example, using the *MC_DeviceNetPDConfig* function block.

	TIP
	The set process data length influences the process data lengths of the polled I/O as well as of the bit-strobe I/O messages. The process data length of the bit-strobe I/O messages can include up to four process data words. If the value for the process data length set via DIP switches is less than four, it will be accepted. If the value set greater than four, the process data length will be automatically limited to four.

Timeout response with polled I/O

The timeout is triggered by the DHF41B option. The timeout interval must be set by the master after the connection has been established. The DeviceNet specification refers to an "expected packet rate" rather than a timeout interval in this case. The expected packet rate is calculated on the basis of the timeout interval using the following formula:

$$t_{Timeout\_inverter} = t_{Timeout\_interval\_polled\_IO} = 4 \times t_{Expected\_packet\_rate\_polled\_IO}$$

The expected packet rate can be set using the connection object class 5, instance 2, attribute 9. The range of values runs from 0 ms to 65535 ms in 5 ms steps.

The expected packet rate for the polled I/O connection is converted into the timeout interval and displayed in the device as timeout interval in index 8606 in the bus diagnostics in the parameter tree.

This timeout interval is retained in the device when the polled I/O connection is disconnected, and the device switches to timeout status after the timeout interval has elapsed.

The timeout interval must not be altered using the PLC Editor or the IEC program because it can only be activated via the bus.

If a timeout occurs for the polled I/O messages, this connection type enters timeout status. Incoming polled I/O messages are no longer accepted.

The timeout triggers the timeout response defined in the IEC program.

The timeout can be reset via DeviceNet using the reset service of the connection object (class 0x05, instance 0x02, undetermined attribute), by disconnecting the connection, by using the reset service of the identity object (class 0x01, instance 0x01, undetermined attribute), or by restarting the MOVI-PLC® *advanced* DHF41B. The timeout response and further handling or blocking of the state must take place in the IEC program.

Bit-strobe I/O

The SEW fieldbus device profile does not include the bit-strobe I/O messages. The messages represent a DeviceNet-specific process data exchange. The master sends a broadcast message that is 8 bytes (= 64 bits) long. One bit in this message is assigned to each station in accordance with its address. The value of this bit may be 0 or 1, triggering two different responses in the recipient.

Bit value	Meaning	LED BIO
0	Sends back process input data only	Green light
1	Trigger fieldbus timeout reaction and send back process input data	Flashing red

NOTICE

The LED L14 (BIO) on the front of the UFF41B option can be used for distinguishing between the timeout triggered by the bit-strobe telegram and a real timeout in the connection. The LED L14 (BIO) lights up green when bit-strobe messages are received cyclically.

LED L14 (BIO) flashing red means there is a timeout in the bit-strobe connection and no more bit-strobe telegrams are accepted. Each participant that has received this bit-strobe I/O message responds with its current process input data. The length of the process input data corresponds to the process data length for the polled I/O connection. The process input data length can be up to four process data.

The following table shows the data range of the bit-strobe request message which represents the allocation of stations (= station address) to data bits.

Example: For example, the participant with station address (MAC-ID) 16 only processes bit 0 in data byte 2.

Byte offset	7	6	5	4	3	2	1	0
0	ID 7	ID 6	ID 5	ID 4	ID 3	ID 2	ID 1	ID 0
1	ID 15	ID 14	ID 13	ID 12	ID 11	ID 10	ID 9	ID 8
2	ID 23	ID 22	ID 21	ID 20	ID 19	ID 18	ID 17	ID 16
3	ID 31	ID 30	ID 29	ID 28	ID 27	ID 26	ID 25	ID 24
4	ID 39	ID 38	ID 37	ID 36	ID 35	ID 34	ID 33	ID 32
5	ID 47	ID 46	ID 45	ID 44	ID 43	ID 42	ID 41	ID 40
6	ID 55	ID 54	ID 53	ID 52	ID 51	ID 50	ID 49	ID 48
7	ID 63	ID 62	ID 61	ID 60	ID 59	ID 58	ID 57	ID 56

Timeout response with bit-strobe I/O

The timeout is triggered by the DHF41B option. The timeout interval must be set by the master after the connection has been established. The DeviceNet specification refers to an "expected packet rate" rather than a timeout interval in this case. The expected packet rate is calculated on the basis of the timeout interval using the following formula:

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

It can be set using connection object class 5, instance 3, attribute 9. The range of values runs from 0 ms to 65535 ms in 5 ms steps.

If a timeout occurs for the bit-strobe I/O messages, this connection type enters timeout status. Incoming bit-strobe I/O messages are no longer accepted. The timeout is not passed to the MOVI-PLC® *advanced* DHF41B.

The timeout can be reset as follows:

- Via DeviceNet with the reset service of the connection object (class 0x05, instance 0x03, undetermined attribute)
- By disconnecting the connection
- Using the reset service of the identity object (class 0x01, instance 0x01, undetermined attribute)

6.2 The Common Industrial Protocol (CIP)

DeviceNet is integrated into the Common Industrial Protocol (CIP).

In the Common Industrial Protocol, all unit data can be accessed via objects. The objects listed in the following table are integrated in the DHF41B option.

Class [hex]	Name
01	Identity object
03	DeviceNet Object
05	Connection Object
07	Register Object
0F	Parameter Object

6.2.1 CIP object directory

Identity object

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

Class

None of the class attributes are supported.

Instance 1

Attri-bute	Access	Name	Data type	Default value [hex]	Description
1	Get	Vendor ID	UINT	013B	SEW-EURODRIVE GmbH & Co KG
2	Get	Device Type	UINT	0064	Manufacturer-specific type
3	Get	Product Code	UINT	000D	Product no.13: MOVI-PLC® advanced
4	Get	Revision	STRUCT of		Revision of the identity object, depends on firmware version
		Major Revision	USINT		
		Minor Revision	USINT		
5	Get	Status	WORD		→ Table "Coding of attribute 5 status"
6	Get	Serial number	UDINT		Unique serial number
7	Get	Product Name	SHORT_STRING	SEW MOVIPLC ADVANCED	Product name

- Coding of 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	→ Table "Coding of 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
0101	No IO connection established
0110	At least one IO connection active

Supported services

Service code [hex]	Service Name	Instance
05	Reset	X
0E	Get_Attribute_Single	X

DeviceNet object

- The DeviceNet object provides information on the DeviceNet communication interface.
- Class code: 03_{hex}

Class

Attribute	Access	Name	Data type	Default value [hex]	Description
1	Get	Revision	UINT	0002	Revision 2

Instance 1

Attribute	Access	Name	Description
1	Get	MAC ID	Depending on DIP switch (0 - 63)
2	Get	Baud rate	Depending on DIP switch (0 - 2)
3	Get	BOI	
4	Get/Set	Bus-off counter	Error counter of the physical CAN interface (0 - 255)
5	Get	Allocation information	
6	Get	MAC-ID switch changed	Information as to whether DIP switch settings vary from MAC-ID
7	Get	Baud rate switch changed	Information as to whether DIP switch settings vary from baud rate
8	Get	MAC-ID switch value	DIP switch setting for MAC ID
9	Get	Baud rate switch value	Actual DIP switch settings for baud rate

Supported services

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

Connection object

- The process and parameter data connections are defined in the connection object.
- Class code: 05_{hex}

Class

None of the class attributes are supported.

Instance	Communication
1	Explicit message
2	Polled I/O
3	Bit-Strobe I/O

Instance 1 ... 3

Attribute	Access	Name
1	Get	State
2	Get	Instance type
3	Get	Transport Class trigger
4	Get	Produce connection ID
5	Get	Consume connection ID
6	Get	Initial com characteristics
7	Get	Produced connection size
8	Get	Consumed connection size
9	Get/Set	Expected packet rate
12	Get	Watchdog time-out action
13	Get	Produced connection path len
14	Get	Produced connection path
15	Get	Consumed connection path len
16	Get	Consumed connection path
17	Get	Production inhibit time

Supported services

Service code [hex]	Service Name	Instance
0x05	Reset	X
0x0E	Get_Attribute_Single	X
0x10	Set_Attribute_Single	X

Register object

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

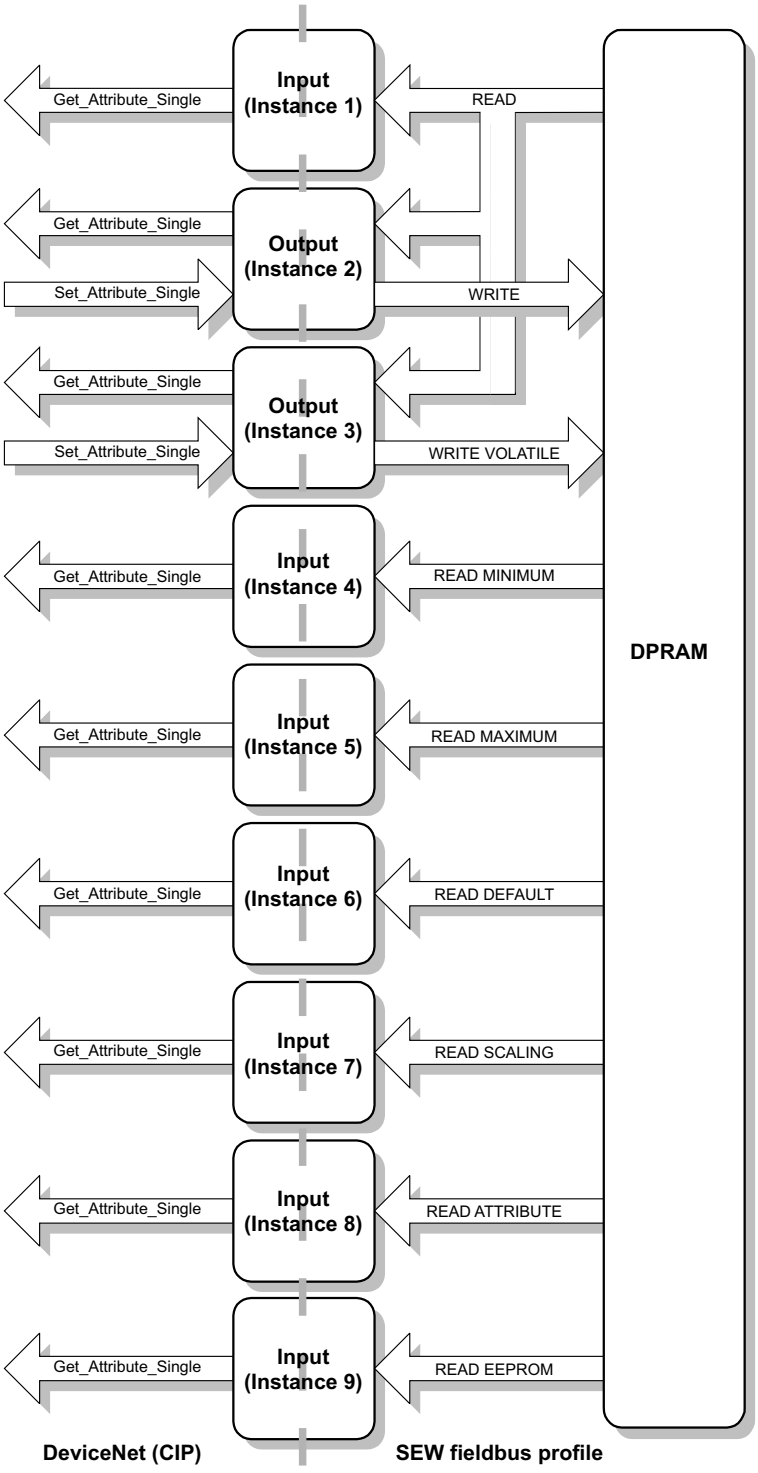
Class

None of the class attributes are supported.

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



62367AEN

Figure 1: Description of the parameter channel

Instance 1 - 9

Attribute	Access	Name	Data type	Default value [hex]	Description
1	Get	Bad Flag	BOOL	00	0 = good / 1 = bad
2	Get	Direction	BOOL	00 01	00 = Input register 01 = 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.

Name	Data type	Description		
Index	UINT	SEW unit index		
Data	UDINT	Data (32 bit)		
Subindex	BYTE	SEW unit subindex		
Reserved	BYTE	Reserved (must be "0")		
Subaddress 1	BYTE	0 Parameter of the MOVI-PLC® itself	1 ...	e.g. SBus address of units connected to the SBus of MOVI-PLC®
Subchannel 1	BYTE	0	3	Lower-level bus system, e.g. SBus 1
Subaddress 2	BYTE	Reserved (must be '0')		
Subchannel 2	BYTE	Reserved (must be '0')		

The subchannels and subaddresses apply depending on the lower-level bus system from MOVI-PLC® *advanced* DHF41B to the drives.

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

Subchannel 1	Interface	Value range subaddress 1
0	MOVI-PLC® itself	0
1	Inverter via DPRAM if in MDX B	0
2	EtherCAT X36	0 - 99 (the EtherCAT address is calculated from: Sub address 1 + 1001)
3	SBus1 (X33 and X26)	1 - 63
4	SBus2 (X32)	1 - 63
5	RS485_1 (X34:1/3/5 and X24)	0 - 99
6	RS485_2 (X34:2/4/6)	0 - 99

Supported services

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

Parameter object

- The fieldbus parameters of the MOVI-PLC® *advanced* DHF41B can be addressed directly via the instance with the parameter object.
- In exceptional cases, you can also use the parameter object to access SEW parameters.
- Class code: $0F_{hex}$

Class

Attribute	Access	Name	Data type	Default value [hex]	Description
2	Get	Max Instance	UINT	0085	Maximum instance = 133
8	Get	Parameter Class Descriptor	UINT	0009	Bit 0: Supports parameter instances Bit 3: Parameters are saved permanently
9	Get	Configura-tion Assem-bly Interface	UINT	0000	Configuration assembly is not supported.

Instance 1 - 133

Instances 1 - 133 provide access to the fieldbus parameters.

Attribute	Access	Name	Data type	Default value [hex]	Description
1	Set/Get	Parameter	UINT		Parameter to be read or written (see section "Fieldbus parameters MOVI-PLC® *advanced* DHF41B")
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

Fieldbus parameters MOVI-PLC® advanced DHF41B

Instance	Access	Group	Name	Meaning
1	Get/Set	Device parameter	PD configuration	Process data configuration
2	Get		Timeout time	Timeout interval
3	Get		Fieldbus type	DeviceNet
4	Get		Baud rate	Baud rate via DIP switches
5	Get		Station address	MAC-ID via DIP switches
6 - 69	Get	PO monitor	PO1 setpoint ... PO64 setpoint	Monitor of the process output data words
70 - 133	Get	PI monitor	PI1 actual value ... PI64 actual value	Monitor of the process input data words

TIP

The data format for these instances deviates from the SEW fieldbus profile to meet the DeviceNet specification.

Supported services

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

6.3 Return codes of the parameterization via explicit messages

SEW-specific return codes

The return codes that SEW units issue in case of incorrect parameterization are independent of the fieldbus. However, the return codes are sent back in a different format when using DeviceNet. The following table shows the data format for a parameter response telegram.

	Byte offset			
	0	**1**	**2**	**3**
Function	MAC ID	Service code [=94hex]	General Error Code	Additional code
Example	01_{hex}	94_{hex}	$1F_{hex}$	10_{hex}

- MAC-ID is the DeviceNet address
- The *Service code* of an error telegram is always 94_{hex}
- The *general error code* of an manufacturer-specific return code is always $1F_{hex}$
- The additional code is described in the table in the "Additional code" section.
- *General Error Codes D0hex* and *D1hex* signal protocol-specific errors to MOVILINK®, such as incorrect address information (see section "MOVILINK®-specific return codes").

The table shows the proprietary error 10_{hex} = *Illegal parameter index.*

Return codes from DeviceNet

DeviceNet-specific return codes are sent in the error message if the data format is not maintained during the transfer or if a service is performed which has not been implemented. The coding of these return codes is described in the DeviceNet specification (see section "General Error Codes").

Timeout of explicit messages

The timeout is triggered by the DHF41B option. The timeout interval must be set by the master after the connection has been established. The DeviceNet specification refers to an "expected packet rate" rather than a timeout interval in this case. The expected packet rate is calculated on the basis of the timeout interval using the following formula:

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

It can be set using connection object class 5, instance 1, attribute 9. The range of values runs from 0 ms to 65535 ms in 5 ms steps.

If a timeout is triggered for the explicit messages, this connection type for the explicit messages is automatically dropped provided that the polled I/O or bit-strobe connections are not in the ESTABLISHED state. This is the default setting of DeviceNet. The connection for explicit messages must be re-established to communicate with these messages again. The timeout is **not** passed to the MOVI-PLC® *advanced* DHF41B.

General error code

DeviceNet-specific error codes

General error code (hex)	Error name	Description
00 - 01		Reserved for DeviceNet
02	Resource unavailable	The source required for performing the service is unavailable
03 - 07		Reserved for DeviceNet
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		Reserved for DeviceNet
0B	Already in requested mode/state	The selected object is already in the requested mode/state
0C	Object state conflict	The selected object cannot perform the service in its current status
0D		Reserved for DeviceNet
0E	Attribute not settable	It is not possible to access the selected object for writing.
0F	Privilege violation	Violation of access right
10	Device state conflict	The current status of the device makes it impossible to perform the required service
11	Reply data too large	The length of the transmitted data is longer than the size of the receive buffer
12		Reserved for DeviceNet
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		Reserved for DeviceNet
18	No stored attribute data	The requested data have not been stored previously
19	Store operation failure	The data could not be stored because an error occurred while saving them
1A - 1E		Reserved for DeviceNet
1F	Vendor specific error	Vendor specific error (see "SEW Fieldbus Device 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 - CF	Future extensions	Reserved by DeviceNet for additional definitions
D0 - DF	Reserved for Object Class and service errors	Use this area if an occurring error cannot be entered in one of the error groups listed above.

MOVILINK®-specific return codes

The following table lists the MOVILINK® protocol-specific error messages and their coding in the DeviceNet response telegram.

General Error Code	Additional code	Description	Corresponds to	
			MOVILINK® Error Code	MOVILINK® Aditional Code
0xD0	0xF0	Unknown error	0x05	0x00
	0xF1	Illegal Service		0x01
	0xF2	No Response		0x02
	0xF3	Different Address		0x03
	0xF4	Different Type		0x04
	0xF5	Different Index		0x05
	0xF6	Different Service		0x06
	0xF7	Different Channel		0x07
	0xF8	Different Block		0x08
	0xF9	No Scope Data		0x09
	0xFA	Illegal Length		0x0A
	0xFB	Illegal Address		0x0B
	0xFC	Illegal Pointer		0x0C
	0xFD	Not enough memory		0x0D
	0xFE	System Error		0x0E
0xD1	0xF0	Communication does not exist		0x0F
	0xF1	Communication not initialized		0x10
	0xF2	Mouse conflict		0x11
	0xF3	Illegal Bus		0x12
	0xF4	FCS Error		0x13
	0xF5	PB Init		0x14
	0xF6	SBUS - Illegal fragment count		0x15
	0xF7	SBUS - Illegal fragment type		0x16
	0xF8	Access denied		0x17
	0xF9 - FE	Not used		

Additional code

The additional code contains SEW-specific return codes for faulty parameterization of the drive inverter.

General Error Code	Additional code[1)]	Description
0x1F	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

1) These error codes correspond to the MOVILINK® additional codes in error class 0x08.

6.4 Terms and definitions

Term	Description
Allocate	Provides a service for setting up a connection.
Attribute	Attribute of an object class or instance. Describes the characteristics of the object class or instance in more detail.
BIO - Bit-Strobe I/O	All stations can be addressed with a broadcast message. The addressed stations respond with the process input data.
Class	DeviceNet object class
Device-Net scanner	Plug-in module for the Allen Bradley PLC which connects the PLC fieldbus to the peripheral devices.
DUP-MAC check	Duplicate MAC-ID test
Explicit message body	Includes the class no., instance no., attribute no. and the data.
Explicit message	Parameter data message; assists in addressing the DeviceNet objects.
Get_Attribute_Single	Read service for a parameter.
Instance	Instance of an object class. Divides the object classes into additional subgroups.
MAC ID	Media Access Control Identifier: node address of the device.
M-File	Provides the data range between the PLC and the scanner module.
Mod/Net	Module/network
Node ID	Node address = MAC-ID
PIO - Polled I/O	Process data channel of DeviceNet; allows process output data to be sent and process input data to be received.
Release	Provides a service for setting up a connection.
Reset	Provides a service for resetting an error.
Rung	SLC500 program line
Service	Service performed via bus, e.g. read service, write service, etc.
Set_Attribute_Single	Write service for a parameter.

7 Error Diagnostics for Operation on the DeviceNet Fieldbus

7.1 Diagnostic procedures

The diagnostic procedures described in the following section demonstrate the fault analysis methods for the following problems:

- MOVI-PLC® *advanced* DHF41B does not operate on DeviceNet
- MOVI-PLC® *advanced* DHF41B cannot be controlled with the DeviceNet master

For detailed information about programming MOVI-PLC® *advanced* DHF41B, refer to the "MOVI-PLC® *advanced* DH..41B" manual.

Step 1: Check the status LED and status display of the DeviceNet scanner

See documentation of the DeviceNet scanner.

Step 2: Check the status LEDs of DHF41B and DIP switch S2

The explanation of the different LEDs can be found in section 4. The following table shows the corresponding unit states and their causes. An "X" indicates that the state of the respective LED is not relevant.

LED				DHF41B	
L16 MOD/NET	**L15 (PIO)**	**L14 (BIO)**	**L13 (BUS FAULT)**	**Status**	**Cause**
Off	Off	Off	Off	Off	No voltage supply of DHF41B, e.g. via X26. DIP switch S2 is not set to the DeviceNet fieldbus interface (see chapter 4).
Off	Yellow	Off	Off	Booting	During boot up and internal synchronization with MOVI-PLC®
Off	Flashing red	X	Off	Baud rate invalid	Invalid baud rate setting via DIP switches
Off	Flashing green	Flashing green	Yellow	No power via X30	Voltage supply via X30D not connected / switched on
Off	Flashing green	Flashing green	Flashing red	Error pas-sive	Wrong baud rate or no other DeviceNet node connected
Red	Red	Red	Off	DUP-MAC error	Address (MAC-ID) is assigned twice in the network
Flashing green	Off	Off	X	Operational	DHF41B active on the bus but without connection to the master (scanner)
Flashing red	Flashing red	X	X	Timeout	Timeout of the PIO connection to the master
Green	Green	X	X	Connected	DHF41B active on the bus with active PIO connection to the master
Flashing red	Green	X	X	Module error	DHF41B with active PIO connection and active error of DHF41B

Step 3: Fault diagnostics

When DHF41B is in status "Connected" or "Module error", the data exchange between master (scanner) and slave (DHF41B) is active. If it is still not possible to transmit data to the IEC application of the MOVI-PLC® *advanced* DHF41B via DeviceNet, the following steps should help you to diagnose the error.

A Are the correct values for the process data words displayed in the PLC Editor?

If yes, continue with F.

B Is bit 0 in DeviceNet control register of the PLC set to "1" to activate process data exchange?

C Are the process data words copied to the right offset in the Local I/O tag of the DeviceNet 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 DHF41B, refer to the documentation of the PLC manufacturer for support.

F Are the process data words being used correctly in the IEC program?

G Was the communication interface configured and activated properly in the IEC application using the MC_DeviceNetPDConfig function block (see chapter "Settings in MOVI-PLC® *advanced* DHF41B")?

H If the cycle time for exchanging process data is longer than expected, calculate the bus load.

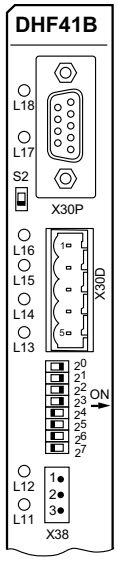
Example:

64 process data words from and to a DeviceNet slave are transmitted in ca. 11 ms at a baud rate of 500 kBaud. When operating 2 units with 64 process data words each, the shortest possible cycle time is about twice as long: ca. 22 ms. Halving the baud rate means doubling the cycle time.

8 Assembly and Installation Notes on the PROFIBUS DP-V1 Fieldbus

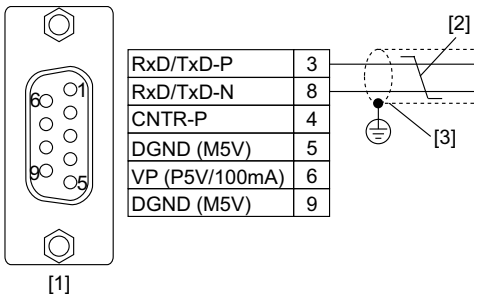
8.1 Connecting MOVI-PLC® advanced DHF41B to a PROFIBUS network

The following sections describe the terminals, DIP switches, and LEDs relevant for PROFIBUS operation.

Front view MOVI-PLC® *advanced* DHF41B controller	Designation	LED DIP switch Terminal		Function
 63598AXX	**LED**	**LED 18** **LED 17** **LED 12** **LED 11**	Run PROFIBUS Fault PROFIBUS - -	Status of PROFIBUS communication Status of PROFIBUS bus electronics Reserved Reserved
	Connector X30P: **PROFIBUS** **(Sub-D9)**	**X30P:9** **X30P:8** **X30P:7** **X30P:6** **X30P:5** **X30P:4** **X30P:3** **X30P:2** **X30P:1**	GND (M5V) RxD/TxD-N N.C. VP (P5V/100 mA) GND (M5V) CNTR-P RxD/TxD-P N.C. N.C.	Reference potential for PROFIBUS Signal receive transmit negative Terminal unassigned DC+5 V potential for bus terminator Reference potential for PROFIBUS PROFIBUS control signal for repeater Signal receive transmit positive Terminal unassigned Terminal unassigned
	DIP switch S2 **Switching between** **PROFIBUS and DeviceNet**	**S2**	Top Bottom	Fieldbus interface PROFIBUS (X30P) active Fieldbus interface DeviceNet (X30P) active
	For PROFIBUS mode: DIP switch for setting the PROFIBUS station address	**2^0** **2^1** **2^2** **2^3** **2^4** **2^5** **2^6**		Significance: 1 Significance: 2 Significance: 4 Significance: 8 Significance: 16 Significance: 32 Significance: 64
	Connector X38: **CAN for safety-relevant communication** **(plug-in terminals)**	**X38:1** **X38:2** **X38:3**		Reserved Reserved Reserved

8.2 Connecting PROFIBUS (connector X30P)

Connection to the PROFIBUS system is made using a 9-pin sub D plug according to IEC 61158. The T-bus connection must be made using a plug with the corresponding configuration. The following figure shows the PROFIBUS connector that is connected to X30P of MOVI-PLC® *advanced* DHF41B.



61766AXX

[1] 9-pin D-sub connector

[2] Signal line, twisted

[3] Conductive connection over a large area is necessary between plug housing and the shield

Connecting MOVI-PLC®/ PROFIBUS

As a rule, the MOVI-PLC® *advanced* DHF41B control card is connected to the PROFIBUS system using a shielded twisted-pair cable. Observe the maximum supported transmission rate when selecting the bus connector.

The twisted-pair cable is connected to the PROFIBUS connector at pin 3 (RxD/TxD-P) and pin 8 (RxD/TxD-N). Communication takes place using these two pins. The RS485 signals RxD/TxD-P and RxD/TxD-N must all be connected to the same contacts in all PROFIBUS stations. Else, the bus components cannot communicate via the bus medium.

The PROFIBUS interface sends a TTL control signal for a repeater or fiber optic adapter (reference = pin 9) via pin 4 (CNTR-P).

Baud rates greater than 1.5 MBaud

MOVI-PLC® *advanced* DHF41B with baud rates > 1.5 MBaud can only be operated with special 12 MBaud PROFIBUS connectors.

Bus termination

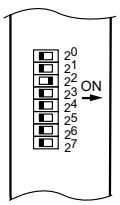
If MOVI-PLC® *advanced* DHF41B is located at the start or end of a PROFIBUS segment and if there is only one PROFIBUS cable connected to MOVI-PLC® *advanced* DHF41B, you must use a connector with an integrated bus terminating resistor.

Switch on the bus terminating resistors for this PROFIBUS connector.

Setting the station address

The PROFIBUS station address is set using DIP switches 2^0 - 2^6 on MOVI-PLC® *advanced* DHF41B.

The MOVI-PLC® *advanced* DHF41B controller supports the address range 0 - 125.



The default setting for the station address is 4:

$2^0 \rightarrow$ Significance: $1 \times 0 = 0$
$2^1 \rightarrow$ Significance: $2 \times 0 = 0$
$2^2 \rightarrow$ Significance: $4 \times 1 = 4$
$2^3 \rightarrow$ Significance: $8 \times 0 = 0$
$2^4 \rightarrow$ Significance: $16 \times 0 = 0$
$2^5 \rightarrow$ Significance: $32 \times 0 = 0$
$2^6 \rightarrow$ Significance: $64 \times 0 = 0$

Any change made to the PROFIBUS station address during operation does not take effect immediately. You must restart the inverter with the MOVI-PLC® *advanced* DHF41B control card for the changes to take effect (power + 24 V off/on).

8.3 Status LED of the DHF41B option

The DHF41B option card is equipped with four two-color LEDs for diagnostic of the PROFIBUS system. They indicate the current status of the DHF41B and the PROFIBUS system. The unit status corresponding to the status of the LED is described in chapter "Error diagnostics".

LED L17 (FAULT PROFIBUS)

LED **L17 (FAULT** PROFIBUS) indicates that communication via the PROFIBUS interface is working properly.

Status of the L17 LED	Diagnostics	Remedy
Off	• MOVI-PLC® *advanced* DHF41B exchanges data with the PROFIBUS-DP master (data exchange status).	-
Red	• Connection to the DP master has failed. • MOVI-PLC® *advanced* DHF41B cannot detect the PROFIBUS baud rate. • Bus interruption has occurred. • PROFIBUS-DP master not in operation.	• Check the PROFIBUS connection on the unit. • Check project planning of the PROFIBUS DP master. • Check all the cables in the PROFIBUS network.
Flashing red (1 Hz)	• MOVI-PLC® *advanced* DHF41B recognizes the baud rate. However, the DP-Master does not address the MOVI-PLC® *advanced* DHF41B control card. • MOVI-PLC® *advanced* DHF41B was either not configured in the DP master or was configured incorrectly.	• Check and correct the PROFIBUS station address set in MOVI-PLC® *advanced* DHF41B and in the configuration software of the DP master. • Check and correct the configuration of the DP master. • Use the GSD file **SEW_6007.GSD** with the designation *MOVI-PLC* for configuration.

LED L18 (RUN PROFIBUS)

LED **L18 (RUN PROFIBUS)** indicates the proper functioning of the PROFIBUS electronics (hardware).

Status of the L18 LED	Diagnostics	Remedy
Green	• PROFIBUS hardware OK.	-
Flashing green (1 Hz)	• The PROFIBUS station address set on the DIP switches exceeds 125. If the PROFIBUS station address is set to a value higher than 125, MOVI-PLC® *advanced* DHF41B will use PROFIBUS station address 4.	1. Check and correct the PROFIBUS station address on the DIP switches. 2. Switch on all drive inverters again. The modified PROFIBUS address will only take effect after a restart.
Orange	• The DHF41B option is being initialized.	-

9 Configuration and Startup on the PROFIBUS DP-V1 Fieldbus

9.1 Configuring a PROFIBUS DP master

You need a GSD file to configure a PROFIBUS DP master for controlling MOVI-PLC® *advanced* DHF41B.

	**TIP**
	The current version of the GSD file for controlling MOVI-PLC® *advanced* DHF41B is available on the SEW website (http://www.sew-eurodrive.de) in the "Software" section.

GSD file for PROFIBUS DP/DP-V1

The **GSD file SEW_6007.GSD** corresponds to GSD revision 4. The device master data files standardized by the PROFIBUS user organization can be read by all PROFIBUS DP masters.

Configuration tool	DP master	File name
All DP configuration tools to IEC 61158	For DP master standard	SEW_6007.GSD
Siemens S7 hardware configuration	For all S7 DP masters	

	**TIP**
	Do not change or expand entries in the GSD file! SEW assumes no liability for malfunctions that occur in MOVI-PLC® or connected inverters if a GSD file has been modified!

General project planning procedure

Proceed as follows to configure the MOVI-PLC® *advanced* DHF41B control card with PROFIBUS DP interface:

1. Install (copy) the GSD file according to the requirements of your configuration software (see manuals of your configuration software or the section "Installing the GSD file in STEP7", below). Once the file has been installed properly, the device appears next to the slave stations with the designation *MOVI-PLC*.
2. For configuration, add the MOVI-PLC® *advanced* DHF41B control card under the name *MOVI-PLC* to the PROFIBUS structure and assign the PROFIBUS station address.
3. Select the process data configuration required for your application (see section "DP Configurations").
4. Enter the I/O or peripheral addresses for the configured data widths.

After configuration, you can start PROFIBUS DP. The red *Fault PROFIBUS* LED indicates the status of the configuration (OFF = configuration OK).

Installing the GSD file in STEP7

Proceed as follows to install the GSD file in STEP7:

1. Start the Simatic Manager.
2. Open an existing project and start the hardware configuration.
3. Close the project window in the HW Config. A new file version cannot be installed when the project window is open.
4. In the menu, click on [Extras] / [Install new GSD...] and select the new GSD file with the name SEW_6007.GSD.

The software installs the GSD file and the associated bitmap files in the STEP7 system.

The SEW drive is available under the following path in the hardware catalog: PROFIBUS DP

+--Additional PERIPHERAL UNITS

+--Drives

+---SEW

+--DPV1

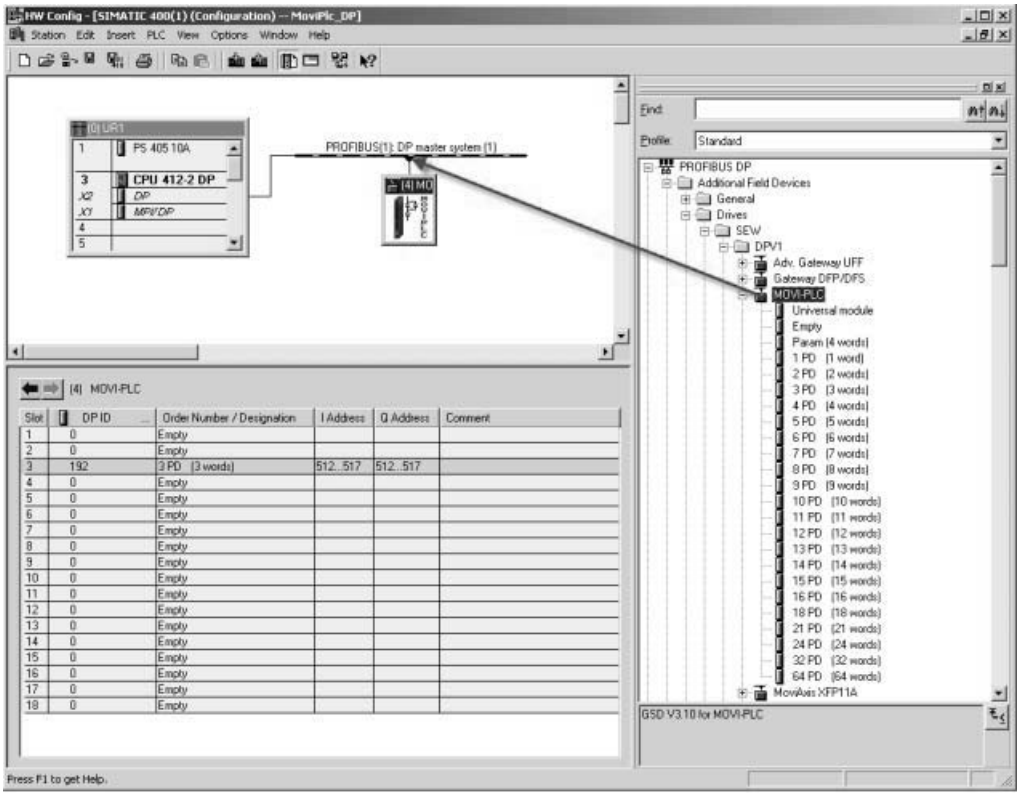
+---MOVI-PLC

The installation of the new GSD file is now complete.

Configuration with STEP7

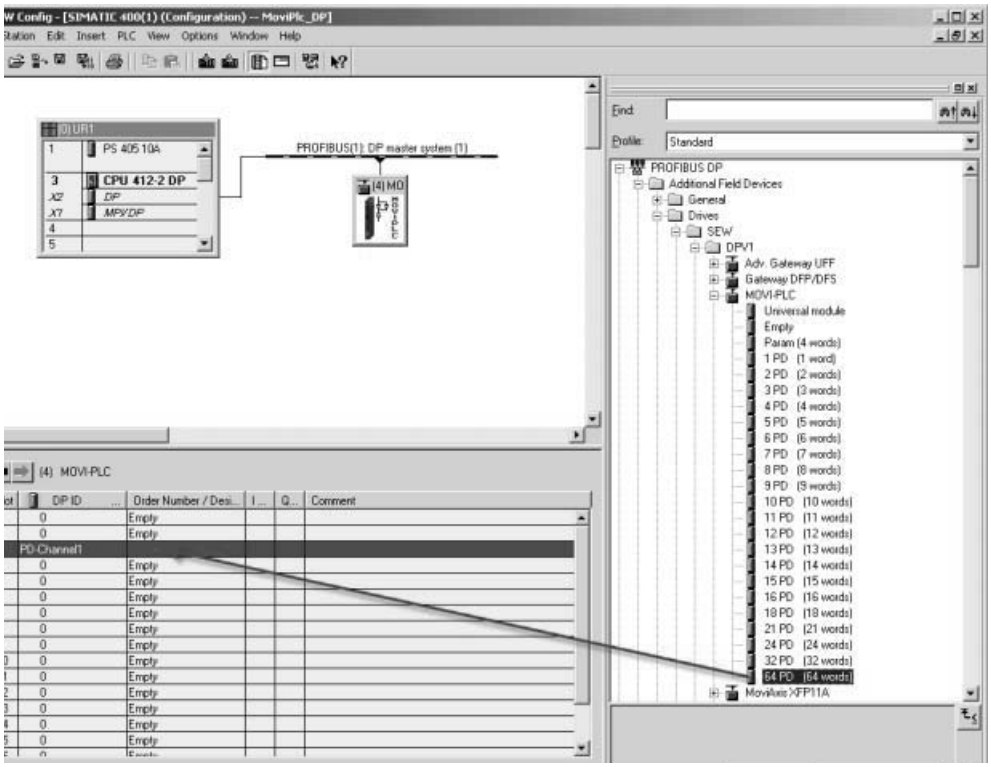
Proceed as follows to configure the MOVI-PLC® *advanced* DHF41B control card with PROFIBUS DP interface:

1. Add the interface module with the name "MOVI-PLC" to the PROFIBUS structure using the drag and drop function and enter the station address (see following figure).



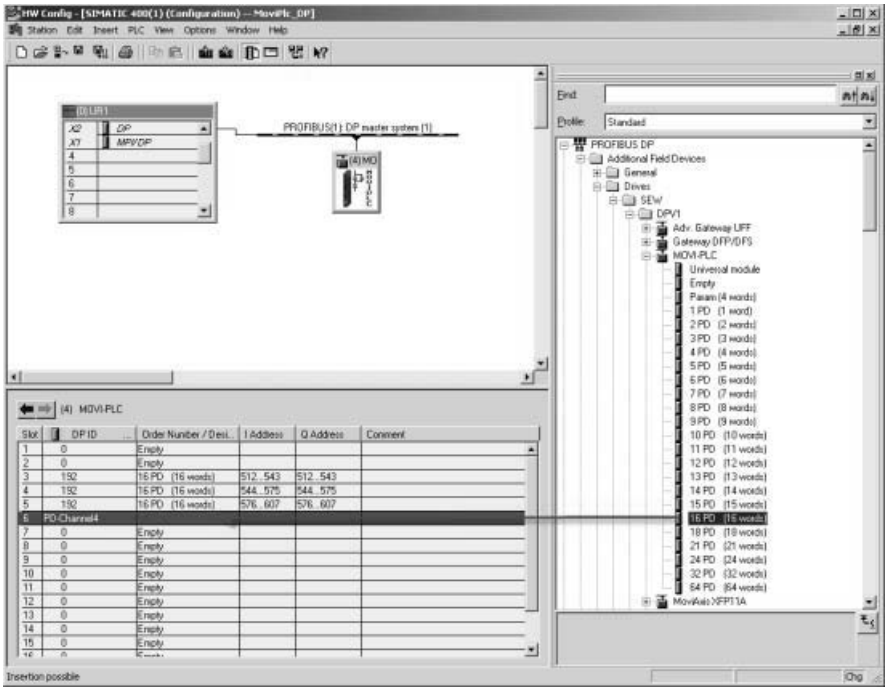
12052AXX

2. The MOVI-PLC® *advanced* DHF41B control card is now preconfigured with the configuration 3PD. To change the PD configuration, you have to delete the 3 PD module in slot 3. Next, add another PD module (e.g. the maximum configuration 64 PD) from the "MOVI-PLC" folder at slot 3 using the drag & drop function (see following figure).



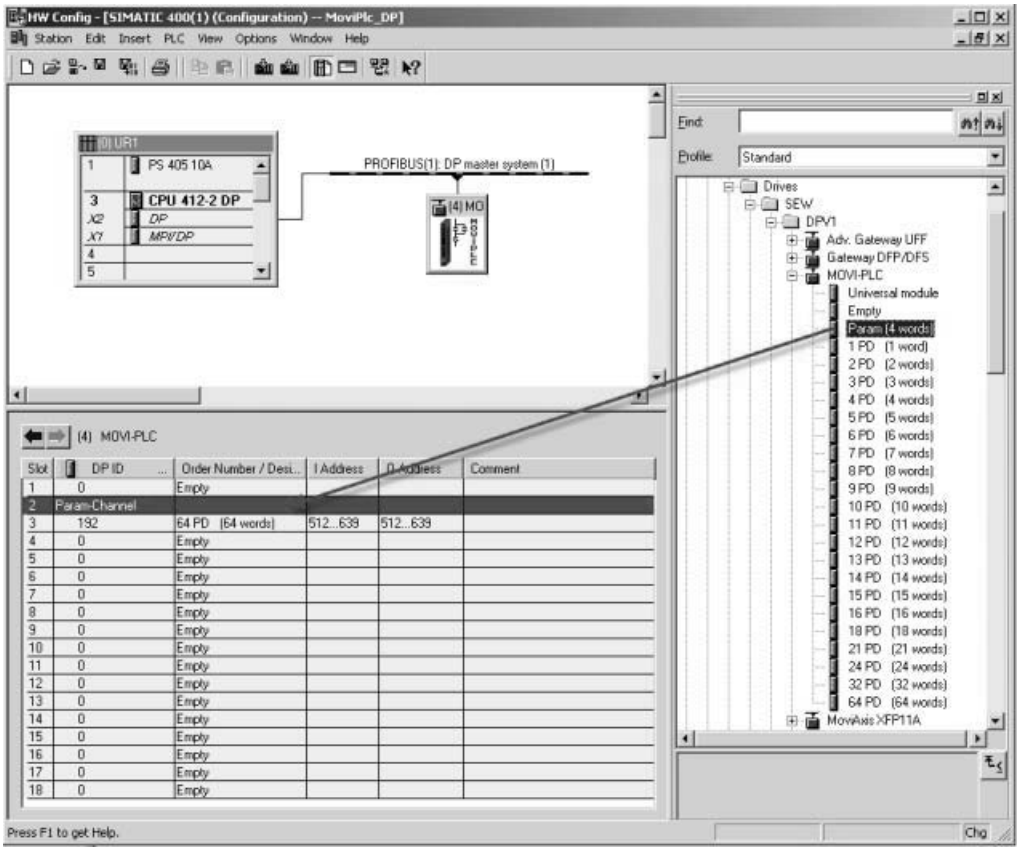
12053AXX

Slots 4 to 18 can be configured in the same way. In the following figure, the maximum configuration 64 PD is distributed among 4 slots (mapping 64 data words in smaller peripheral areas).



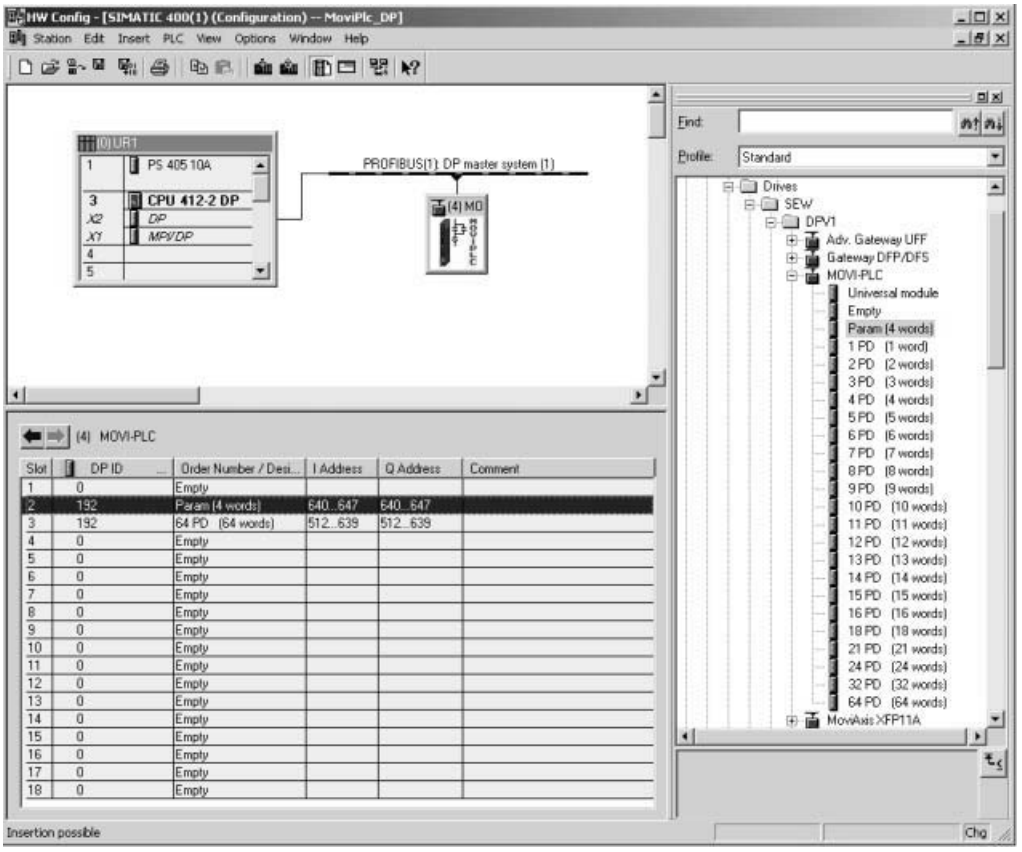
12054AXX

3. Optionally, you can configure a MOVILINK® parameter channel in the cyclic process data (see following figure). To do so, delete the "Empty" module from slot 2 and replace it with the module "Param (4 words)" using the drag and drop function.



12055AXX

4. Enter the I/O or peripheral addresses for the configured data widths in the "I Address" [1] and "Q Address" [2] columns.



12056AXX

DP configuration

To enable the MOVI-PLC® *advanced* DHF41B control card to support the type and amount of input and output data to be transmitted, the DP master on the MOVI-PLC® *advanced* DHF41B control card must transmit the corresponding DP configuration. The configuration telegram comprises the DP configurations for slots 1 to 18.

You can:

- Control the MOVI-PLC® *advanced* DHF41B control card using process data
- Read or write parameters using the parameter channel

The following tables contain additional information on possible DP configurations.

- The "Parameter data/Process data configuration" column displays the name of the configuration. These names also appear in a selection list in the configuration software for the DP master.
- The "DP configurations" column shows the configuration data that is sent to the MOVI-PLC® *advanced* DHF41B control card when the link to the PROFIBUS DP system is being established.

Slot 1:

Parameter data Configuration	Meaning / notes	DP configurations
Empty	Reserved	0x00

Slot 2:

Parameter data Configuration	Meaning / notes	DP configurations
Empty	Reserved	0x00
Param (4words)	MOVILINK® parameter channel configured	0xC0, 0x87, 0x87

Slots 4 to 18:

Process data Configuration	Meaning / notes	DP configurations
1 PD	Process data exchange via 1 process data word	0xC0, 0xC0, 0xC0
2 PD	Process data exchange via 2 process data words	0xC0, 0xC1, 0xC1
3 PD	Process data exchange via 3 process data words	0xC0, 0xC2, 0xC2
4 PD	Process data exchange via 4 process data words	0xC0, 0xC3, 0xC3
5 PD	Process data exchange via 5 process data words	0xC0, 0xC4, 0xC4
6 PD	Process data exchange via 6 process data words	0xC0, 0xC5, 0xC5
7 PD	Process data exchange via 7 process data words	0xC0, 0xC6, 0xC6
8 PD	Process data exchange via 8 process data words	0xC0, 0xC7, 0xC7
9 PD	Process data exchange via 9 process data words	0xC0, 0xC8, 0xC8
10 PD	Process data exchange via 10 process data words	0xC0, 0xC9, 0xC9
11 PD	Process data exchange via 11 process data words	0xC0, 0xCA, 0xCA
12 PD	Process data exchange via 12 process data words	0xC0, 0xC7, 0xC7
13 PD	Process data exchange via 13 process data words	0xC0, 0xCC, 0xCC
14 PD	Process data exchange via 14 process data words	0xC0, 0xCD, 0xCD
15 PD	Process data exchange via 15 process data words	0xC0, 0xCE, 0xCE
16 PD	Process data exchange via 16 process data words	0xC0, 0xCF, 0xCF
32 PD	Process data exchange via 32 process data words	0xC0, 0xDF, 0xDF
64 PD	Process data exchange via 64 process data words	0xC0, 0xFF, 0xFF

Configuration example

Slot 1: Empty

Slot 2: Param (4 words)

Slot 3: 10 PD

Configuration telegram sent to the MOVI-PLC® *advanced* DHF41B control card: 0x00 0xC0 0xC87 0x87 0xC0 0xC9 0xC9

	TIP
	The MOVI-PLC® *advanced* DHF41B control card does not support the "Special identifier formats" coding! Only use the setting "Integrity over entire length" for data transmission.

Data integrity

Integral data is data that always has to be transmitted between the higher-level controller and the MOVI-PLC® *advanced* DHF41B control card as one block and must never be transmitted separately.

Data integrity is particularly important for transmitting position values or complete positioning tasks. The reason that data integrity is so important is that data which is not transmitted integrally could come from different program cycles of the higher-level controller, which would lead to undefined values being transmitted to MOVI-PLC® *advanced* DHF41B controller.

For PROFIBUS DP, data communication between the higher-level control and the MOVI-PLC® *advanced* DHF41B control card is carried out with the setting "Data integrity over entire length".

10 PROFIBUS DP-V1 Operating Characteristics

This section describes the principle characteristics of the MOVI-PLC® *advanced* DHF41B control card when operated on the PROFIBUS-DP system.

10.1 Process data exchange with MOVI-PLC® advanced DHF41B

MOVI-PLC® *advanced* DHF41B is controlled via the process data channel which is up to 64 I/O words in length. These process data words are mapped in the I/O or peripheral area of MOVI-PLC® *advanced* DHF41B, for example, when a higher-level PLC is used as the DP master. As a result, the process data words can be addressed in the usual manner.

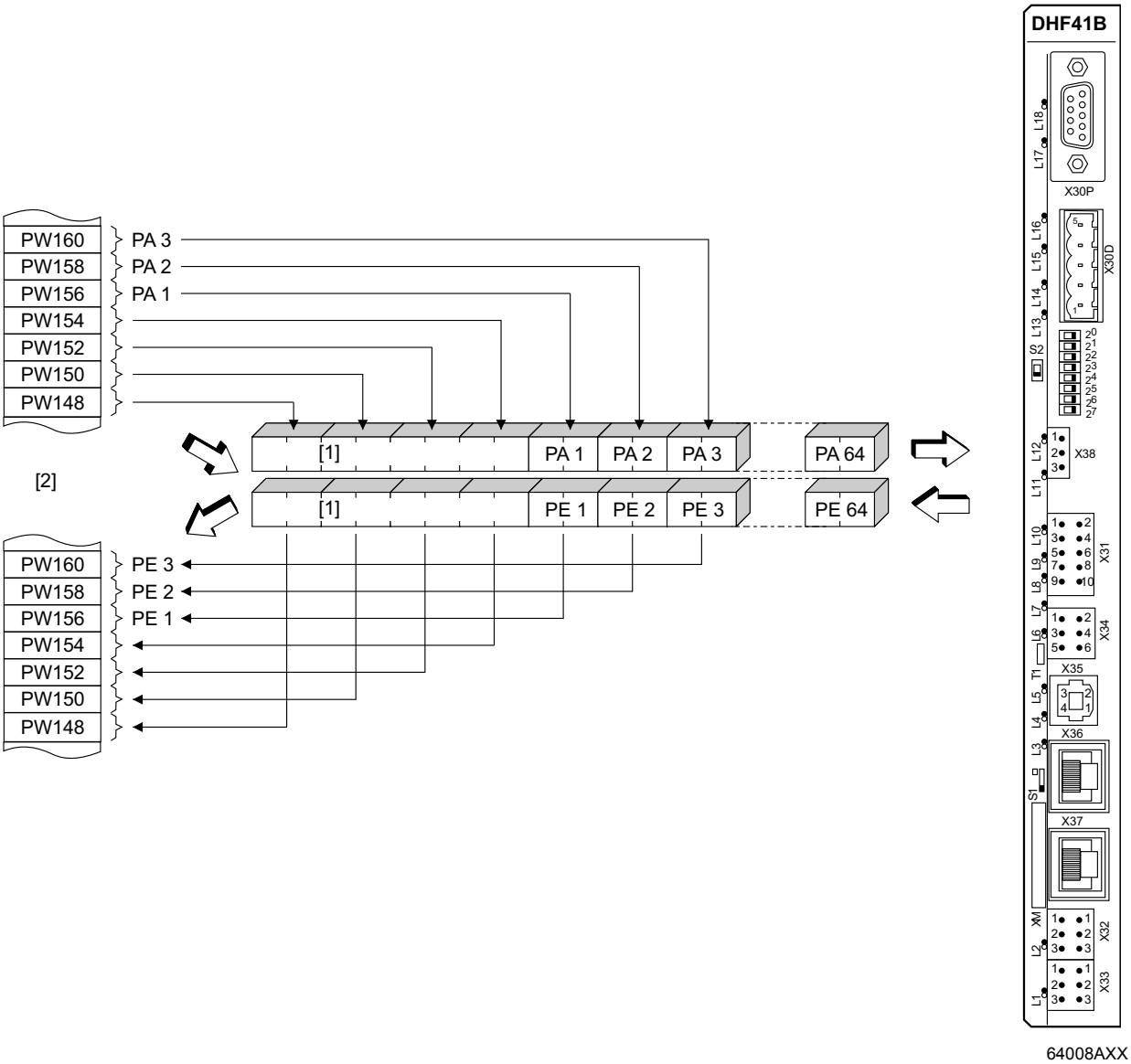


Figure 2: Mapping PROFIBUS data in the PLC address range

[1] 8 byte MOVILINK® parameter channel

[2] Address range of the higher-level PLC

PI1 - PI64 Process input data

PO1 - PO64 Process output data

Control example for Simatic S7

Process data exchange with MOVI-PLC® *advanced* DHF41B is controlled using Simatic S7 in accordance with the selected process data configuration either directly using load and transfer commands or by means of special system functions, *SFC 14 DPRD_DAT* and *SFC15 DPWR_DAT*.

STEP7 example program

In this example, MOVI-PLC® *advanced* DHF41B is configured with the process data configuration *10 PD* on 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.

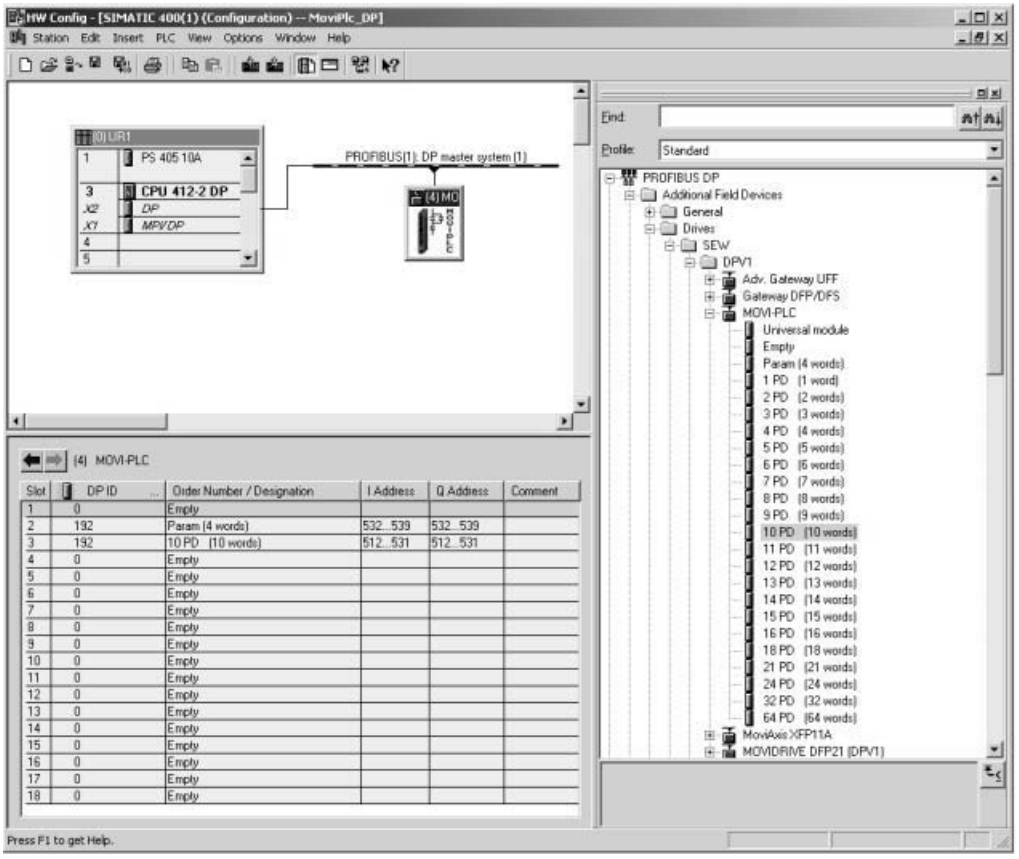
Refer to the online help for STEP7 for further information about the system functions.

```
//Start of cyclical program processing in OB1
BEGIN
NETWORK
TITLE = Copy PI data from the DHF41B 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 DHF41B 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 DHF41B 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.

The following figure shows the corresponding configuration for the MOVI-PLC® *advanced* DHF41B control card in the hardware configuration of STEP7 (see section "DP configurations", page 57).



12067AXX

10.2 PROFIBUS-DP timeout

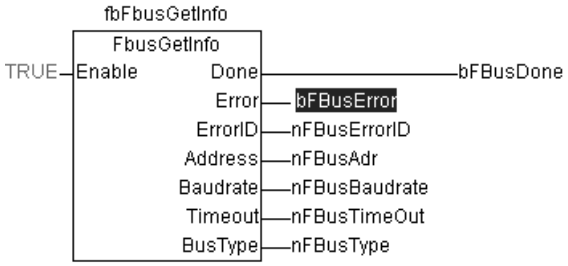
The response monitoring time on the MOVI-PLC® *advanced* DHF41B control card elapses (if configured in the DP master) if data transmission via the PROFIBUS-DP system is interrupted or disrupted. The *Fault PROFIBUS* LED lights up to indicate that no new user data is being received.

The cyclically executable *ProfibusGetInfo* function blockin the MPLCInterface_Profibus library indicates this PROFIBUS timeout. The fault response can be programmed explicitly. The application flow can be influenced accordingly.

10.3 Settings in MOVI-PLC® advanced DFHF41B

The creation of IEC programs is described in detail in the "MOVI-PLC®" manual. This section only describes the fieldbus-specific characteristics.

10.3.1 Status of the PROFINET fieldbus interface



12046AXX

The *FbusGetInfo* function block makes the status and some display parameters of the fieldbus interface available for the IEC program and diagnostics.

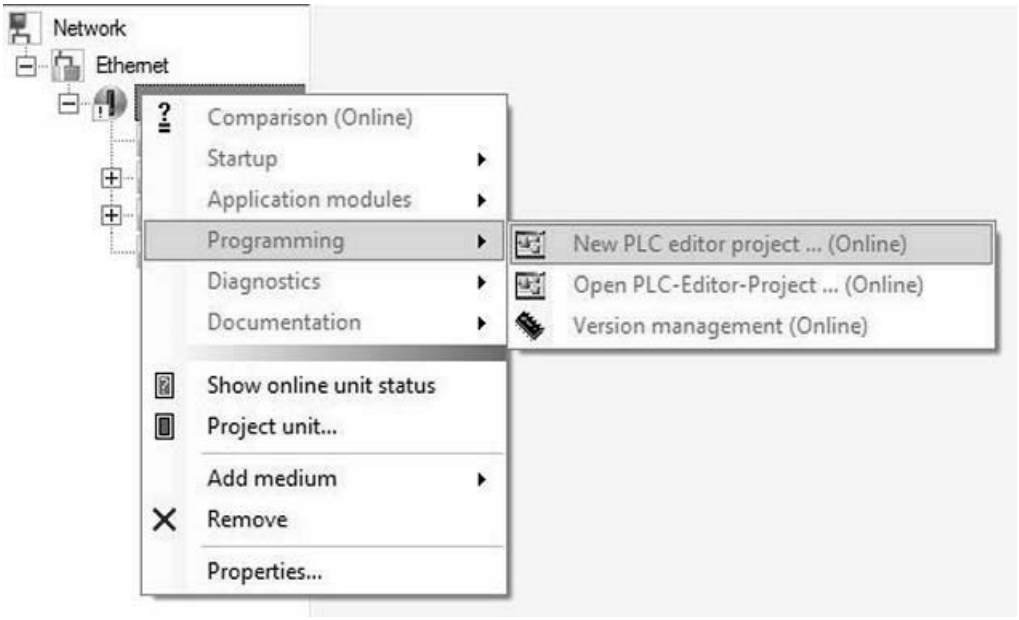
If there is no communication with the fieldbus master, the output *Error* is set to *TRUE*. During an active fieldbus connection, the output *Done* is set to *TRUE*, and the outputs *Address*, *Baud rate*, *Timeout* and *Bus type* show the respective parameters as they were set via the DIP switches of the DHF41B option or via the PLC.

Checking the process data communication

The MOVI-PLC® *advanced* DHF41B can now read the actual values and write setpoints. The process data should correspond to the values displayed in the PLC Editor or in the diagnostics plug-in of the active IEC program in MOVITOOLS® MotionStudio.

If there is no IEC program in MOVI-PLC®, you can create one as follows:

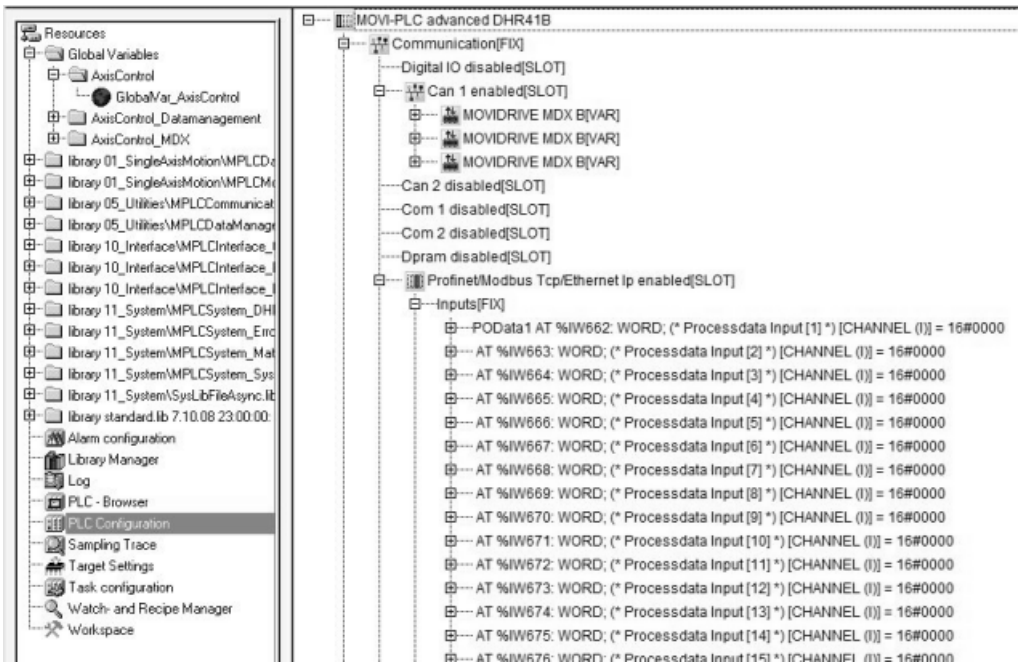
- Open the context menu of the PLC in MOVITOOLS® MotionStudio and run the project wizard "Create new PLC Editor project" (see following figure).



12049AXX

- Use the wizard to create a new AxisControl project and upload it to MOVI-PLC® *advanced* DHF41B using the menu item "Online login".

- Start the loaded program via the menu item "Online start". You can now monitor the uploaded process data under "Resources Control configuration " (PLC configuration). (See following figure).



12050AXX

10.4 Parameter setting via PROFIBUS-DP

In the PROFIBUS-DP system, the parameters are accessed via the 8-byte MOVILINK® parameter channel. This parameter channel offers extra parameter services in addition to the conventional *Read* and *Write* services.

Structure of the 8-byte MOVILINK® parameter channel

PROFIBUS-DP enables access to the MOVI-PLC® *advanced* DHF41B control card parameters via the "parameter process data object" (PPO). This PPO is transmitted cyclically and in addition to the process data channel [2], contains a parameter channel [1], which can be used to exchange acyclical parameter values (see following figure).

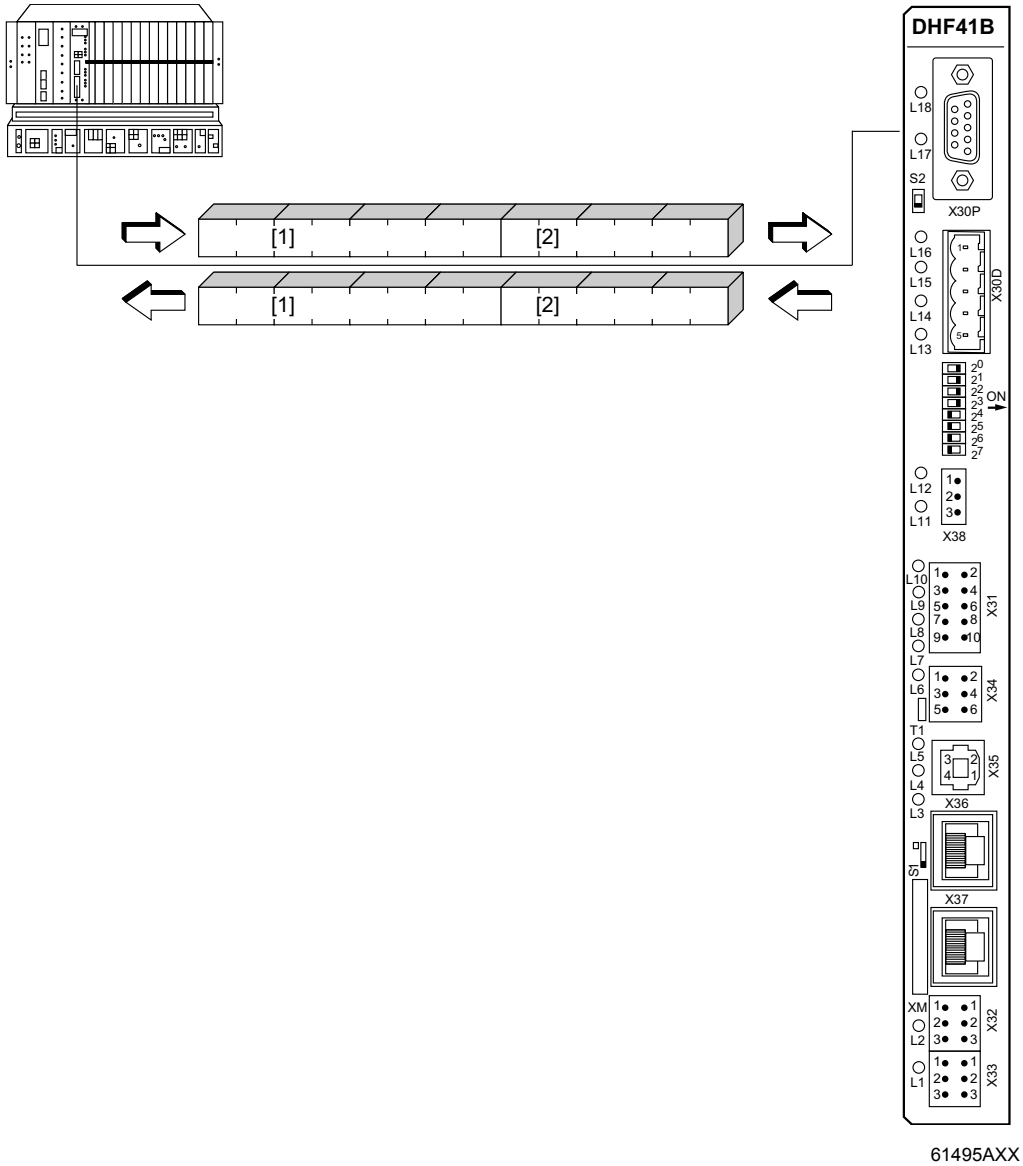


Figure 3: Communication via PROFIBUS-DP

The following table shows the structure of the 8-byte MOVILINK® parameter channel. Its basic structure is as follows:

- One management byte
- One reserved byte
- Two index bytes
- Four data bytes

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Manage-ment	Subindex	Index high	Index low	MSB data	Data	Data	LSB data
		Parameter index		4-byte data			

Management of the 8-byte MOVILINK® parameter channel

The entire procedure for setting parameters is coordinated using management byte 0. This byte provides important service parameters such as service identifier, data length, version and status of the service performed.

The following table shows the structure of the 8-byte MOVILINK® parameter channel.

7 / MSB	6	5	4	3	2	1	0 / LSB
				Service identifier 0000 = No service 0001 = Read parameter 0010 = Write parameter 0011 = Write parameter volatile 0100 = Read minimum 0101 = Read maximum 0110 = Read default 0111 = Read scale 1000 = Read attribute			
		Data length 00 = 1 byte 01 = 2 bytes 10 = 3 bytes 11 = 4 bytes (must be set)					
	Handshake bit Must be changed on every new task in cyclical transmission.						
Status bit 0 = No error during execution of service 1 = Error during execution of service							

- Bits 0, 1, 2 and 3 contain the service identifier. These bits determine which service is to be executed.
- Bits 4 and 5 specify the data length in bytes for the write service. The data length should be set to 4 bytes for the MOVI-PLC® *advanced* DHF41B control card.
- Bit 6 is used as the handshake bit between the higher-level controller and the MOVI-PLC® *advanced* DHF41B control card. Bit 6 triggers the implementation of the transmitted service in the control card. With PROFIBUS-DP, the parameter channel is transmitted cyclically with the process data. For this reason, the execution of the service in MOVI-PLC® *advanced* DHF41B must be triggered by edge control using handshake bit 6. For this purpose, the value of this bit is toggled for each new service to be executed. The MOVI-PLC® *advanced* DHF41B control card uses handshake bit 6 to signal whether the service was executed or not. The service was executed if the handshake bit received in the control is identical with the transmitted handshake bit.
- Status bit 7 indicates whether the service was carried out properly or if errors occurred.

Index addressing

Byte 2: Index high and byte 3: Index low determines the parameter that is to be read or written via the fieldbus system. The parameters of MOVI-PLC® *advanced* DHF41B are addressed using a standard index regardless of the fieldbus system that is connected.

Byte 1 is the subindex.

Data range

As shown in the following table, the data is contained in byte 4 through byte 7 of the parameter channel. This means up to 4 bytes of data can be transmitted per service. The data is always entered with right-justification; that is, byte 7 contains the least significant data byte (Data LSB) whereas byte 4 is the most significant data byte (Data MSB).

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Manage-ment	Subindex	Index high	Index low	MSB data	Data	Data	LSB data
				High byte 1	Low byte 1	High byte 2	Low byte 2
				High word		Low word	
				Double word			

Incorrect service execution

The status bit in management byte 0 is set to signal that a service has been executed incorrectly. If the received handshake bit is identical to the transmitted handshake bit, the MOVI-PLC® advanced DHF41B control card has executed the service. If the status bit now signals an error, the error code is entered in the data range of the parameter telegram. Byte 4 ... 7 send back the return code in a structured format (see the chapter "Return codes for parameterization").

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Manage-ment	Subindex	Index high	Index low	Error class	Error code	Add. code high	Add. code low
Status bit = 1: Incorrect service execution							

Reading a parameter with PROFIBUS-DP (Read)

Due to the cyclical transfer of the parameter channel, to execute a read service via the 8-byte MOVILINK® parameter channel, the handshake bit may only be changed if the complete parameter channel has been set up for the specific service. As a result, adhere to the following sequence when reading a parameter:

1. Enter the index of the parameter to be read in byte 2 (index high) and byte 3 (index low).
2. Enter the service identifier for the read service in the management byte (byte 0).
3. Transfer the write service to MOVI-PLC® *advanced* DHF41B by changing the handshake bit.

Since this is a read service, the sent data bytes (bytes 4...7) and the data length (in the management byte) are ignored and do not need to be set.

The MOVI-PLC® *advanced* DHF41B control card now processes the read service and sends the service confirmation back by changing the handshake bit.

7 / MSB	6	5	4	3	2	1	0 / LSB
0	0/1[1]	X[2]	X[2]	0	0	0	1
				Service identifier 0001 = Read parameter			
		Data length Irrelevant for read service					
	Handshake bit Must be changed on every new task in cyclical transmission.						
Status bit 0 = No error during execution of service 1 = Error during execution of service							

1) Bit value is changed

2) Not relevant

The above table shows how a read service is coded in management byte 0. The data length is irrelevant, only the service identifier for the read service must be entered. This service is now activated in the MOVI-PLC® *advanced* DHF41B control card when the handshake bit changes. For example, it would be possible to activate the read service with the management byte coding 01hex or 41hex.

Writing a parameter with PROFIBUS-DP (write)

Due to the cyclical transfer of the parameter channel, to execute a write service via the 8-byte MOVILINK® parameter channel, the handshake bit may only be changed if the complete parameter channel has been set up for the specific service. Observe the following sequence when writing a parameter:

1. Enter the index of the parameter to be written in byte 2 (index high) and byte 3 (index low).
2. Enter the data to be written in bytes 4 through 7.
3. Enter the service identifier and the data length for the write service in the management byte (byte 0).
4. Transfer the write service to MOVI-PLC® *advanced* DHF41B by changing the handshake bit.

MOVI-PLC® *advanced* DHF41B now processes the write service and sends the service confirmation back by changing the handshake bit.

The following table shows how a write service is coded in management byte 0. The data length for all parameters in the MOVI-PLC® advanced DHF41B control card is four bytes. This service is now transferred to the MOVI-PLC® *advanced* DHF41B control card when the handshake bit changes. Consequently, a write service on the MOVI-PLC® *advanced* DHF41B control card always has the management byte coding 32hex or 72hex.

7 / MSB	6	5	4	3	2	1	0 / LSB
0	0/1[1)]	1	1	0	0	1	0
				Service identifier 0010 = Write parameter			
		Data length 11 = 4 bytes					
	Handshake bit Must be changed on every new task in cyclical transmission.						
Status bit 0 = No error during execution of service 1 = Error during execution of service							

1) Bit value is changed

Parameter setting sequence with PROFIBUS DP

Taking the example of the write service, the following table shows the parameter setting sequence between higher-level controller and MOVI-PLC® *advanced* DHF41B control card via PROFIBUS-DP. To simplify the sequence, only the management byte of the parameter channel is shown here.

While the higher-level controller prepares the parameter channel for the write service, the parameter channel is only received and returned by the MOVI-PLC® *advanced* DHF41B control card. The service is not activated until the moment the handshake bit is changed (in this example, when it changes from 0 to 1). The MOVI-PLC® *advanced* DHF41B control card now interrupts the parameter channel and processes the write service. However, it continues to respond to all telegrams with handshake bit = 0.

The executed service is acknowledged with a change of the handshake bit in the response message of MOVI-PLC® *advanced*DHF41B. The higher-level controller now detects that the received handshake bit is the same as the one that was sent. It can now prepare another parameter setting procedure.

Controller	PROFIBUS-DP(V0)			MOVI-PLC® DHF41B control card (slave)
	--	0**0**110010XXX...	→	Parameter channel is received, but not evaluated
	←	0**0**110010XXX...	--	
Parameter channel is prepared for write service				
Handshake bit is changed and the service is transferred to the MOVI-PLC® DHF41B control card	--	0**1**110010XXX...	→	
	←	0**0**110010XXX...	--	
	--	0**1**110010XXX...	→	
	←	0**0**110010XXX...	--	Write service is performed, handshake bit is changed
Service confirmation is received as the send and receive handshake bits are the same again	←	0**1**110010XXX...	--	
	--	0**1**110010XXX...	→	Parameter channel is received, but not evaluated

Parameter data format

When parameters are set via the fieldbus interface, the same parameter coding is used as with the serial RS485 interface or the system bus.

10.5 Return codes for parameter setting

Elements

If parameters are set incorrectly, the MOVI-PLC® *advanced* DHF41B control card returns various return codes to the parameter setting master, providing detailed information on the cause of the error. Generally, these return codes are structured. SEW distinguishes between the elements

- *Error class*
- *Error code*
- *Additional code*

These return codes are described in detail in the Fieldbus Communications Profile manual and are not included in this documentation. However, the following special cases can occur in connection with PROFIBUS:

Error class

The *error class* element provides a more exact classification of the error type. The MOVI-PLC® *advanced* DHF41B control card supports the following error classes in accordance with EN 50170(V2):

Class (hex)	Designation	Meaning
1	vfd state	Status error of the virtual field device
2	application reference	Error in application program
3	definition	Definition error
4	resource	Resource error
5	service	Error during execution of service
6	access	Access error
7	ov	Error in the object list
8	other	Other error (→ Additional code)

Error code

The *error code* element provides a means of identifying the cause of the error within the *error class*. It is generated by the communication software of the fieldbus card in the event of an error in communication. For *Error class 8 = Other error*, only *Error code = 0 (Other error code)* is defined. In this case, detailed identification is made using the *additional code*.

Additional code

The *Additional code* contains SEW-specific return codes for errors in the parameter settings for MOVI-PLC® *advanced* DHF41B. These codes are returned to the master under *Error class 8 = Other error*. The following table shows all possible codes for the *Additional code*.

Add. code high (hex)	Add. code low (hex)	Meaning
00	00	No error
00	10	Illegal parameter index
00	11	Function/parameter not implemented
00	12	Read access only
00	13	Parameter lock is active
00	14	Factory setting is active
00	15	Value for parameter too large
00	16	Value for parameter too small
00	17	Reserved
00	18	Error in system software
00	19	Reserved
00	1A	Parameter access only via RS485 interface
00	1B	Parameter is access-protected
00	1C	Reserved
00	1D	Invalid value for parameter
00	1E	Factory setting was activated
00	1F	Reserved
00	20	Reserved

10.6 Special cases

Special return codes

Errors in parameter settings, which cannot be identified either automatically by the application layer of the fieldbus system or by the system software of the MOVI-PLC® *advanced* DHF41B control card, are treated as special cases. The following errors can occur depending on the control card used:

- Incorrect coding of a service via parameter channel
- Incorrect length specification of a service via parameter channel
- Internal communication error

Incorrect service code in the parameter channel

An incorrect code was specified for the management byte or reserved byte during parameter setting via the parameter channel. The following table shows the return code for this special case.

	Code (dec)	Meaning
Error class:	5	Service
Error code:	5	Illegal parameter
Add. code high:	0	-
Add. code low:	0	-

Troubleshooting:

Check bits 0 and 1 in the parameter channel.

Incorrect length specification in parameter channel

A data length other than 4 data bytes was specified in a *read* or write service during parameter setting via the parameter channel. The following table displays the return codes.

	Code (dec)	Meaning
Error class:	6	Access
Error code:	8	Type conflict
Add. code high:	0	-
Add. code low:	0	-

Troubleshooting:

Check bit 4 and bit 5 for the data length in management byte 0 of the parameter channel. Both bits must be set to "1".

Internal communication error

The return code listed in the following table is sent back if an internal communication error has occurred. The parameter service transferred via the fieldbus may not have been performed and should be repeated. If this error persists, switch the MOVI-PLC® *advanced* DHF41B control card off and on again. In this way, the control card is reinitialized.

	Code (dec)	Meaning
Error class:	6	Access
Error code:	2	Hardware fault
Add. code high:	0	-
Add. code low:	0	-

Rectifying the error:

Repeat the *read* or write service. If this error occurs again, disconnect the MOVI-PLC® *advanced* DHF41B control card from the power source and switch the system on again. If the error persists, consult SEW Service.

11 Functions of PROFIBUS DP-V1

This section provides you with information about the PROFIBUS DP-V1 functions.

11.1 Introduction to PROFIBUS DP-V1

This chapter describes the functions and terms used for operating SEW units on PROFIBUS DP-V1. Refer to the PROFIBUS user organization or visit www.profibus.com for detailed technical information on PROFIBUS DP-V1.

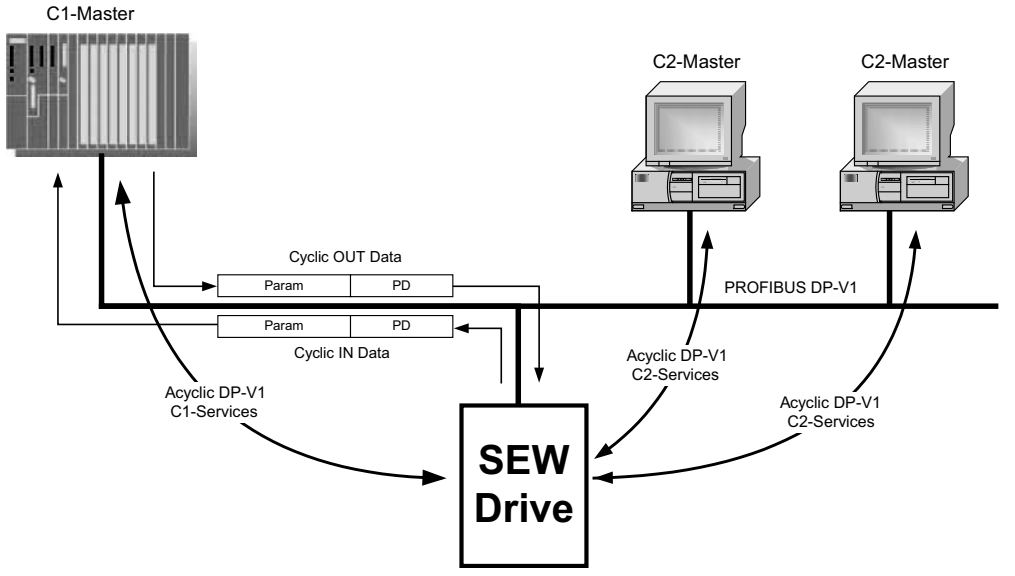
The PROFIBUS DP-V1 specification introduced new acyclical *READ / WRITE* services as part of the PROFIBUS DP-V1 expansions. These acyclical services are inserted into special telegrams during ongoing cyclical bus operation and therefore ensure compatibility between PROFIBUS DP (version 0) and PROFIBUS DP**V1** (**V**ersion **1**).

The acyclical *READ/WRITE* services can be used to exchange larger data quantities between master and slave (inverter) than it would be possible to transfer in the cyclical input or output data using the 8-byte parameter channel, for example. The advantage of the acyclical data exchange via DP-V1 lies in the minimum load on the cyclical bus operation since DP-V1 telegrams are only added to the bus cycle if required.

The DP-V1 parameter channel provides the user with 2 options:

- The higher-level controller can access all the device information of the SEW DP-V1 slaves. This means that cyclical process data and unit settings can be read, stored in the controller and modified in the slave.
- It is also possible to route the service and startup tool MOVITOOLS® MotionStudio via the DP-V1 parameter channel instead of using a proprietary RS485 connection. Once you have installed the MOVITOOLS® MotionStudio software, you can access detailed information in the folder ...\SEW\MOVITOOLS\Fieldbus.

The main features of PROFIBUS DP-V1 are explained below.



11.1.1 Class 1 master (C1 master)

The PROFIBUS DP-V1 network differentiates between various master classes. The C1 master essentially performs the cyclical data exchange with the slaves. A typical C1 master is a control system, such as a PLC, that exchanges cyclical process data with the slave. If the DP-V1 function has been activated via the GSD file, the acyclical connection between C1 master and slave is established automatically when the cyclical connection of the PROFIBUS DP-V1 is being established. Only one C1 master can be operated in a PROFIBUS DP-V1 network.

11.1.2 Class 2 master (C2 master)

The C2 master itself does not perform cyclical data exchange with the slaves. Examples for a typical C2 master are visualization systems or temporary installed programming devices (Notebook / PC). The C2 master uses exclusively acyclic connections for communication with the slaves. The acyclic connections between C2 master and slave are established by the *Initiate* service. The connection is established once the *Initiate* service has been performed successfully. An established connection enables cyclical data exchange with the slaves using *READ* or *WRITE* services. Several C2 masters can be active in a DP-V1 network. The number of C2 connections, established simultaneously for a slave, is determined by the slave. SEW inverters support two parallel C2 connections.

11.1.3 Data sets (DS)

The user data transported via a DP-V1 service are collected in data sets. Each data set is represented uniquely by its length, a slot number and an index. The structure of data set 47 is used for DP-V1 communication with the SEW inverter. This data set is defined as the DP-V1 parameter channel for drives as of V3.1 in the PROFIdrive profile drive engineering of the PROFIBUS user organization. Different procedures for accessing parameter data in the inverter are provided via this parameter channel.

11.1.4 DP-V1 services

The DP-V1 expansions offer new services, which can be used for acyclical data exchange between master and slave. The system distinguishes between the following services:

C1 master	Connection type: MSAC1 (master/slave acyclical C1)
READ	Read data record
WRITE	Write data record

C2 master	Connection type: MSAC2 (master/slave acyclical C2)
INITIATE	Establish C2 connection
ABORT	Disconnect C2 connection
READ	Read data record
WRITE	Write data record

11.1.5 DP-V1 alarm handling

In addition to the acyclical services, the DP-V1 specification also defines extended alarm handling. Alarm handling now distinguishes between different alarm types. As a result, unit-specific diagnostics cannot be evaluated in DP-V1 operation using the "DDLM_SlaveDiag" DP-V0 service. DP-V1 alarm handling has not been defined for drive engineering as an inverter does not usually transfer its status information via cyclical process data communication.

11.2 Features of SEW fieldbus interfaces

The SEW fieldbus interfaces to PROFIBUS DP-V1 have the same communication features for the DP-V1 interface. The drives are usually controlled via a C1 master with cyclical process data in accordance with the DP-V1 standard. The READ and WRITE services give the C1 master access to the parameters of the fieldbus gateway and lower-level stations stations via the DP-V1 C1 channel.

Two additional C2 channels can be connected in parallel to these parameter setting channels. The first C2 master as a visualization device, for example could use these channels to read parameter data, and a second C2 master in the form of a notebook could use them to configure the drive using the MOVITOOLS® MotionStudio software.

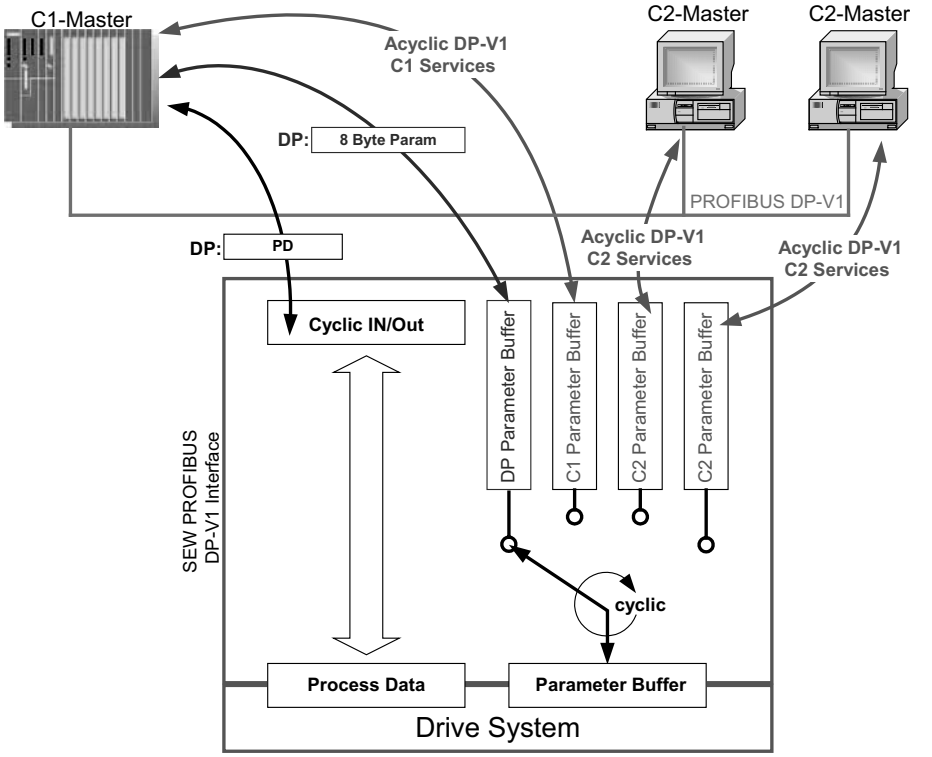


Figure 4: Parameter setting channels for PROFIBUS DP-V1

11.3 Structure of the DP-V1 parameter channel

Generally, the parameter setting of the drives to the PROFIdrive DP-V1 parameter channel of profile version 3.0 is implemented via data set 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 set structure is the same for PROFIdrive and MOVILINK® access.

53125AXX

The following MOVILINK® services are supported:

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

The following PROFIdrive services are supported:

- Reading (request parameter) individual parameters of the type *double word*
- Writing (change parameter) individual parameters of the type *double word*

Field	Data type	Values
Request reference	Unsigned8	0x00 Reserved 0x01 - 0xFF
Request ID	Unsigned8	0x01 Request parameter (PROFIdrive) 0x02 Change parameter (PROFIdrive) 0x40 SEW MOVILINK® service
Response ID	Unsigned8	Response (+): 0x00 Reserved 0x01 Request parameter (+) (PROFIdrive) 0x02 Change parameter (+) (PROFIdrive) **0x40 SEW MOVILINK® service (+)** Response (-): 0x81 Request parameter (-) (PROFIdrive) 0x82 Change parameter (-) (PROFIdrive) **0xC0 SEW MOVILINK® service (-)**
Axis	Unsigned8	0x00 - 0xFF Number of axes 0 - 255
No. of parameters	Unsigned8	0x01 - 0x13 1 - 19 DWORDs (240 DP-V1 data bytes)
Attribute	Unsigned8	0x10 Value **For SEW MOVILINK® (Request ID = 0x40):** **0x00 No service** **0x10 READ parameters** **0x20 WRITE parameter** **0x30 WRITE Parameter volatile** **0x40 ... 0xF0 Reserved**
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 SEW: always 0
Format	Unsigned8	0x43 Double word 0x44 Error
No. of Values	Unsigned8	0x00 - 0xEA Quantity 0 - 234
Error Value	Unsigned16	0x0000 - 0x0064 PROFIdrive error codes 0x0080 + MOVILINK®-Additional Code Low **For SEW MOVILINK® 16 Bit error value**

11.3.1 Parameterization procedure via data set 47

Parameter access takes place with the combination of the DP-V1 services *WRITE* and *READ*. The parameter setting order is transferred to the slave using the *WRITE.req*, followed by slave-internal processing.

The master now sends a *READ.req* to pick up the parameter setting response. The master repeats the *READ.req* if the *READ.res* from the slave is negative. As soon as the parameter processing in the inverter is concluded, it answers with 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 both a C1 and a C2 master.

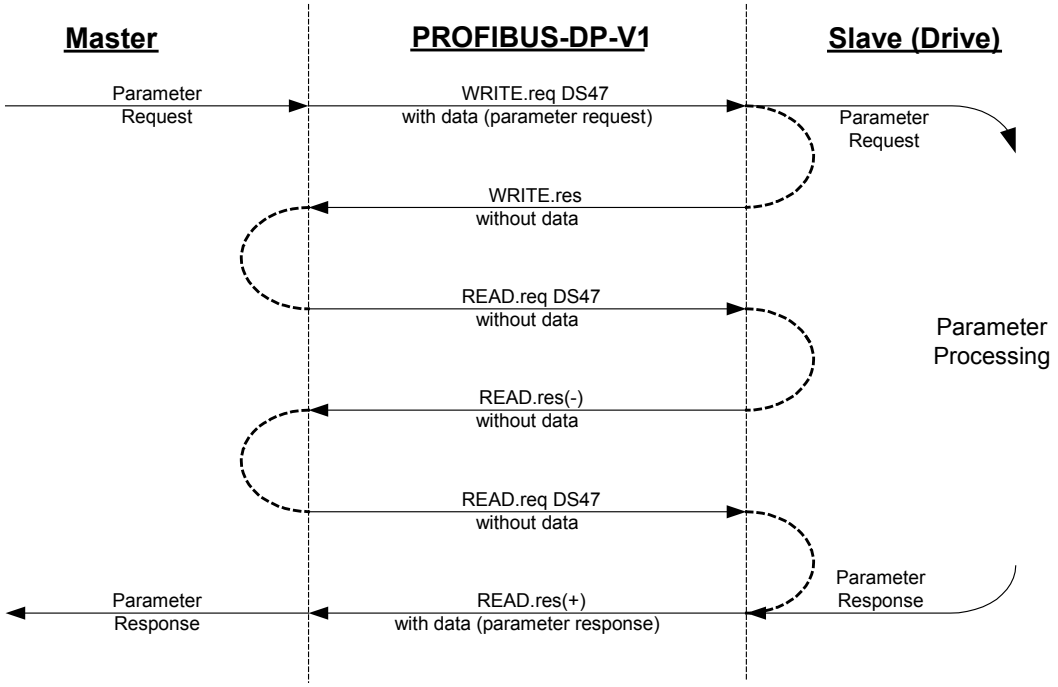
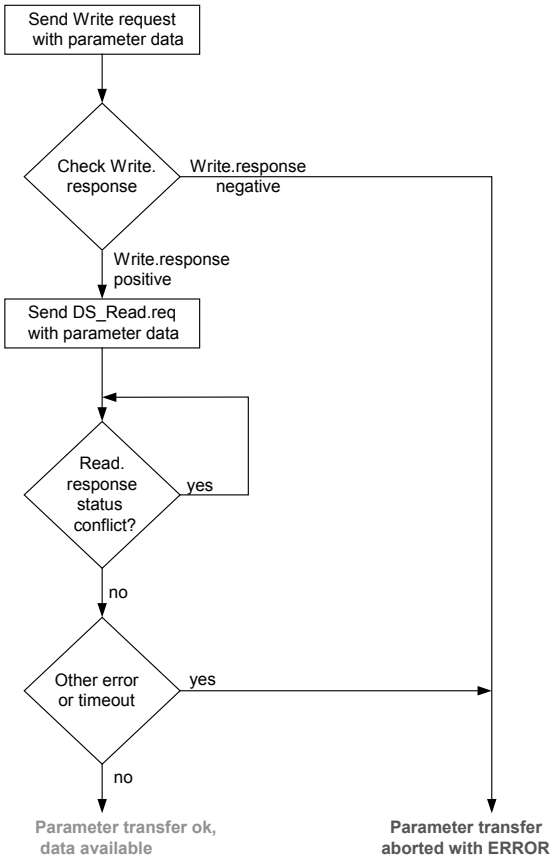


Figure 5: Telegram sequence for parameter access via PROFIBUS DP-V1

11.3.2 DP-V1 master processing sequence

If the bus cycles are very short, the request for the parameter response arrives before the inverter has concluded parameter access in the device. This means that the response data from the inverter is not yet available. In this case, the inverter sends a negative answer with the **Error_Code _1 = 0xB5 (status conflict)** to the DP-V1 level. The DP-V1 master must then repeat the request with the READ.req header until it receives a positive answer from the inverter.



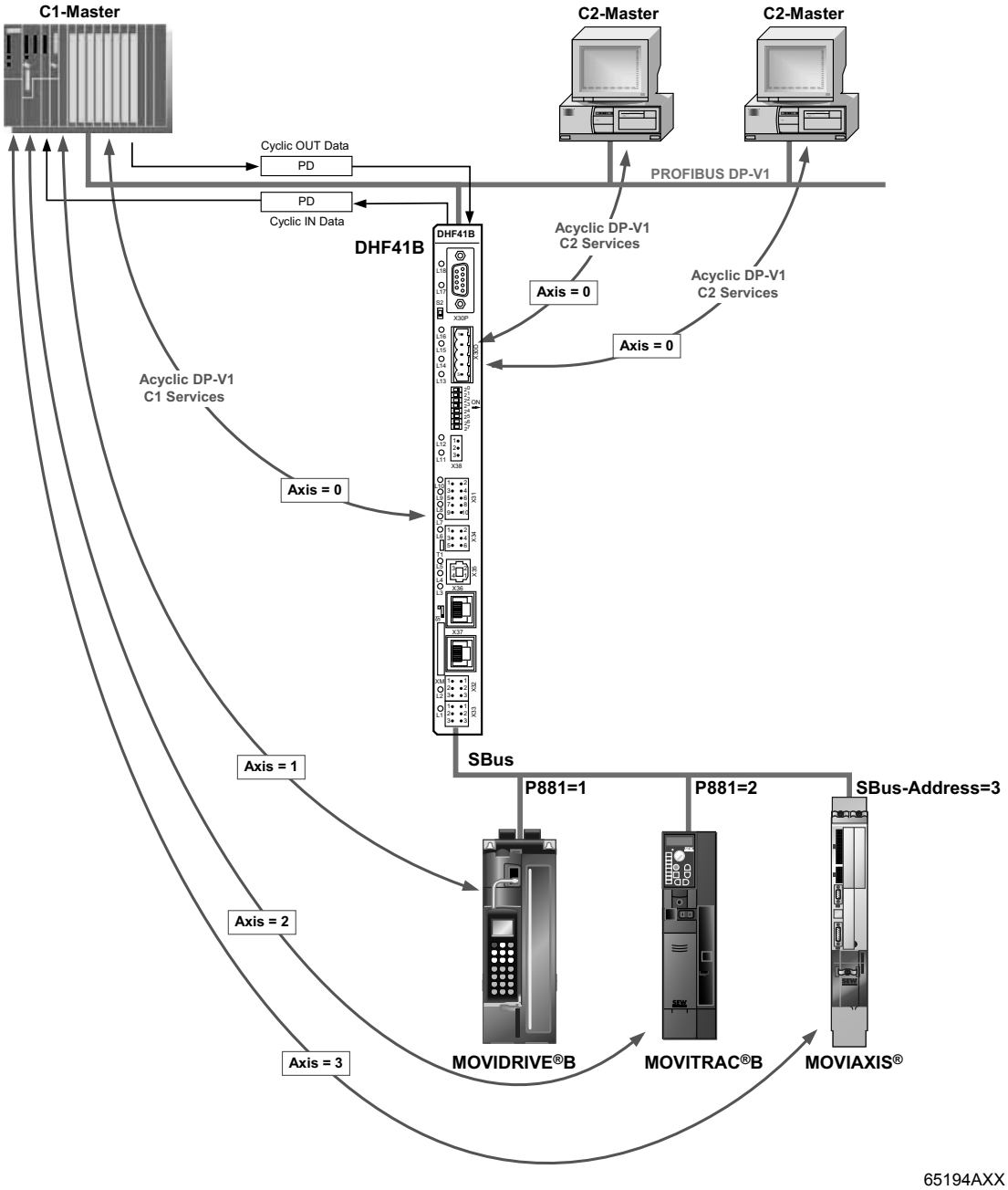
53127AEN

11.3.3 Addressing connected inverters

The structure of the DS47 data set defines an *axis* element. This element is used to reach multi-axis drives that are operated via one PROFIBUS interface. The *axis* element addresses one of the devices connected via the PROFIBUS interface. This mechanism can be used, for example, by the SEW bus modules type DHF, UFF, MOVIFIT®, MQP for MOVIMOT ® or DFP for MOVITRAC® B.

Addressing a MOVIDRIVE® inverter at PROFIBUS DP-V1

Setting *Axis = 0* enables access to the parameters of MOVI-PLC® *advanced* DHF41B. To being able to access slave units connected to the DHF41B option, the setting must be *Axis = SBus address*. SBus address 15 must not be used when engineering via PROFIBUS or parameter services via PROFIBUS.



11.3.4 MOVILINK® parameter requests

The MOVILINK® parameter channel of the SEW inverter is directly mapped in the structure of data set 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 message sequence for data set 47 is used.

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 this element corresponds to the service nibble in the management byte of the DP parameter channel.

Example for reading a parameter via MOVILINK®

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. In the example, the firmware of MOVIDRIVE® B connected to CAN 1 of the DHF41B option is read with SBus address 1.

Sending a parameter request

The following table shows the coding of the user data for the *WRITE.req* service specifying the DP-V1 header. The *WRITE.req* service is used to transfer the parameter setting request to the inverter. The firmware version is read.

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	Request reference	0x01	Individual reference number for the parameter setting request is mirrored in the parameter response.
1	Request ID	0x40	SEW MOVILINK® service
2	SBus address of the unit connected to the gateway	0x01	Axis number; 1 = SBus address 1 at CAN 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.req USER DATA including the DP-V1 header.

Service:	READ.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data set; constant index 47
Length	240	Maximum length of response buffer in the DP-V1 master

Positive MOVILINK® **parameter setting 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 set; constant index 47
Length	10	10 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	0x40	Positive MOVILINK® response
2	Axis	0x01	Mirrored axis number; 1 = 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 MOVILINK®

The following tables show the sequence of the *WRITE* and *READ* services for volatile writing of the value 12345 to IPOS$^{plus®}$ variable H0 (parameter index 11000) as an example. The MOVILINK® service *WRITE Parameter volatile* is used for this purpose. In this example as well, MOVIDRIVE® B with SBus address 1 is connected to the DHF41B option.

Send "WRITE parameter volatile" 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

Byte	Field	Value	Description
0	Request reference	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	Axis number; 1 = SBus address of MDX
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 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.

Query parameter response

The following table shows the coding of the WRITE.req USER DATA including the DP-V1 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 DP master

Positive response to "WRITE Parameter volatile"

Service:	READ.response	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data set; constant index 47
Length	4	4 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	0x40	Positive MOVILINK® response
2	Axis	0x01	Mirrored axis number; 1 = 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:	READ.response	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data set; 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	0xC0	Negative MOVILINK® response
2	Axis	0x01	Mirrored axis number; 1 = 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 DP-V1" on page 85)

MOVILINK® configuration return codes for DP-V1

The following table shows the return codes that are returned by the SEW DP-V1 interface if an error occurs during DP-V1 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)

11.3.5 PROFIdrive parameter orders

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

Request ID:0x01request parameter (PROFIdrive)

Request ID:0x02change parameter (PROFIdrive)

This means there is restricted data access 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.

Example for reading a parameter via PROFIdrive

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 following table shows the coding of the user data for the WRITE.req service specifying the DP-V1 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 set; constant index 47
Length	10	10 byte user data for parameter request

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

Byte	Field	Value	Description
0	Request reference	0x01	Individual reference number for the parameter setting request is mirrored in the parameter response
1	Request ID	0x01	Request parameter (PROFIdrive)
2	Axis	0x01	Axis number; 1 = 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 DP-V1 header.

Service:	READ.request	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data set; constant index 47
Length	240	Maximum length of response buffer in the DP-V1 master

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 set; constant index 47
Length	10	10 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	0x01	Positive response for "Request Parameter"
2	Axis	0x01	Mirrored axis number; 1 = 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 (see section "Example for writing a parameter via MOVILINK®, page 83). The PROFIdrive *Change parameter* service is used for this purpose.

Send "WRITE parameter" request

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

Byte	Field	Value	Description
0	Request reference	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 = 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.

Query parameter response

The following table shows the coding of the WRITE.req user data including the DP-V1 header.

Field	Value	Description
Function_Num		READ.req
Slot_Number	X	Slot_Number not used
Index	47	Index of the data set
Length	240	Maximum length of response buffer in the DP-V1 master

Positive response to "WRITE parameter"

Service:	READ.response	Description
Slot_Number	0	Random, (is not evaluated)
Index	47	Index of the data set; constant index 47
Length	4	4 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	0x02	Positive PROFIdrive response
2	Axis	0x01	Mirrored axis number; 1 = 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 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 set; 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	Mirrored axis number; 1 = 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 DP-V1" on page 85)

PROFIdrive return codes for DP-V1

The following table shows the coding of the error number in the PROFIdrive DP-V1 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 query 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	-

11.4 Configuring a C1 master

A special GSD file *SEW_6007.GSD* is required for configuring a DP-V1 C1 master. This file activates the DP-V1 functions of the DHF41B. The functions of the GSD file and the DHF41B firmware must correspond with one another.

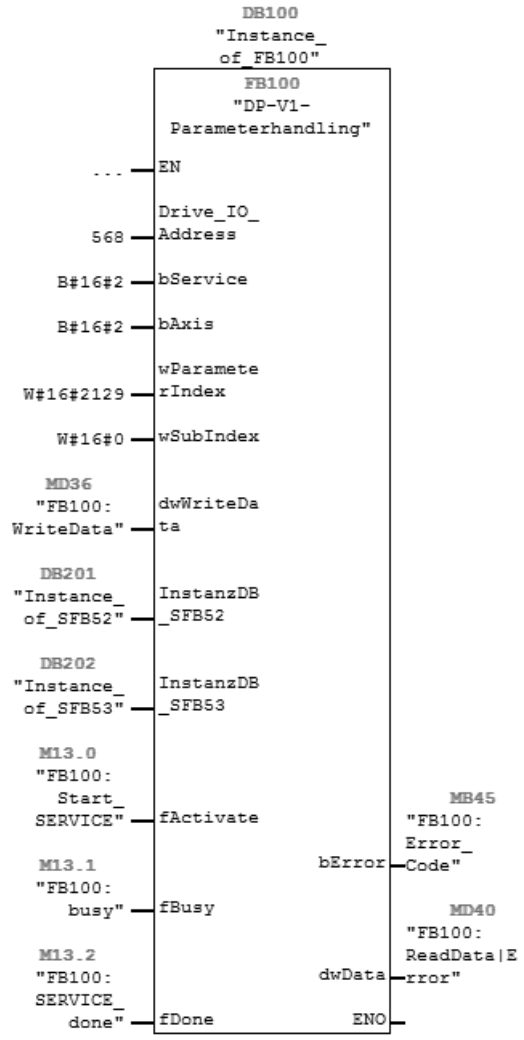
11.4.1 Operating mode (DP-V1 mode)

The DP-V1 operating mode can usually be activated for configuring a C1 master. All DP slaves, which have the DP-V1 functions enabled in their GSD files and which support DP-V1, will then be operated in DP-V1 mode. Standard DP slaves will still run via PROFIBUS DP-V0. This ensures mixed mode for DP-V1 and DP-V0 capable modules. Depending on the master functionality, a DP-V1 capable station, that was configured using the DP-V1 GSD file, can run in the "DP-V0" operating mode.

11.4.2 Example program for SIMATIC S7

	TIPS
	The MOVILINK® parameter channel sample program is available from the SEW homepage (www.sew-eurodrive.de) 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
```

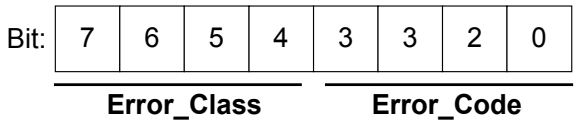
12102AEN

11.4.3 Technical data of DP-V1 for MOVI-PLC® *advanced* DHF41B

GSD file for DP-V1:	SEW_6007.GSD
Module name for project planning:	MOVI-PLC
Number of parallel C2 connections:	2
Supported data set:	Index 47
Supported slot number:	Recommended: 0
Manufacturer code:	10A hex (SEW-EURODRIVE)
Profile ID:	3A
C2 response timeout	1 s
Max. length C1 channel:	240 bytes
Max. length C2 channel:	240 bytes

11.4.4 Error codes of the DP-V1 services

This table shows possible error codes of DP-V1 services that may occur in the event of an error in the communication on DP-V1 telegram level. This table is relevant if you want to write your own parameter assignment block based on the DP-V1 services because the error codes are reported directly back on the telegram level.



Error_Class (from DP-V1 specification)	Error_Class (from DP-V1 specification)	DP-V1 parameter channel
0x0 - 0x9 hex = reserved		
0xA = application	0x0 = read error 0x1 = write error 0x2 = module failure 0x3 to 0x7 = reserved 0x8 = version conflict 0x9 = feature not supported 0xA to 0xF = user specific	
0xB = access	0x0 = invalid index	0xB0 = No data block Index 47 (DB47); parameter requests are not supported
	0x1 = write length error 0x2 = invalid slot 0x3 = type conflict 0x4 = invalid area	
	0x5 = state conflict	0xB5 = Access to DB 47 temporarily not possible due to internal processing status
	0x6 = access denied	
	0x7 = invalid range	0xB7 = WRITE DB 47 with error in the DB 47 header
	0x8 = invalid parameter 0x9 = invalid type 0xA to 0xF = user specific	
0xC = resource	0x0 = read constraint conflict 0x1 = write constraint conflict 0x2 = resource busy 0x3 = resource unavailable 0x4 - 0x7 = reserved 0x8 - 0xF = user specific	
0xD - 0xF = user specific		

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:

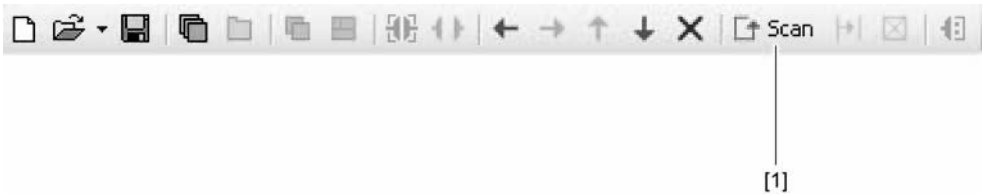
   "Start\Program\SEW\MOVITOOLS MotionStudio\MOVITOOLS MotionStudio"
2. Create a project with name and storage location.

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.

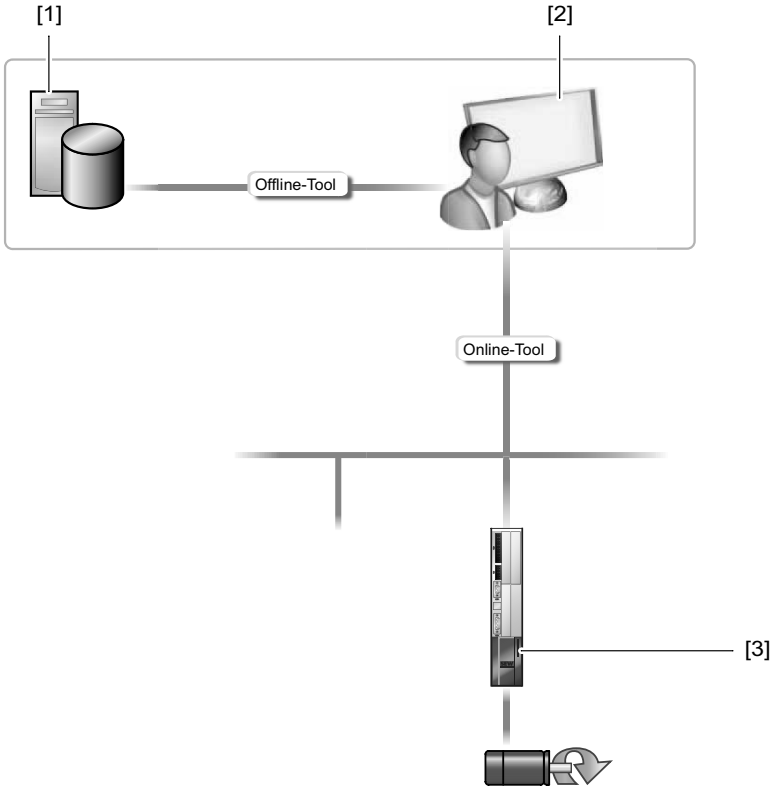
12.3 Communication mode

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.

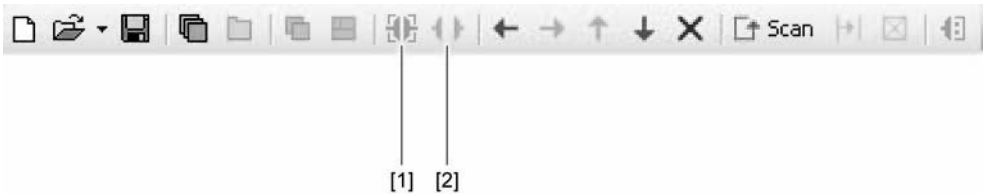
	TIP
	The "online" communication mode is "NOT" a response message which informs you that you are currently connected to the unit or that your unit is ready for communication. • Should you require this feedback, observe section "Setting the cyclical accessibility test" in the online help (or the manual) 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.



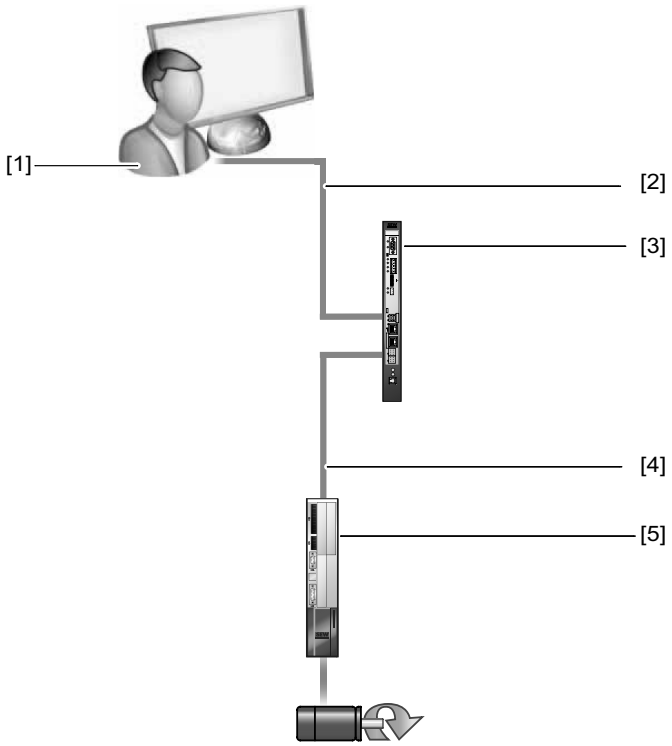
64337AXX

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:

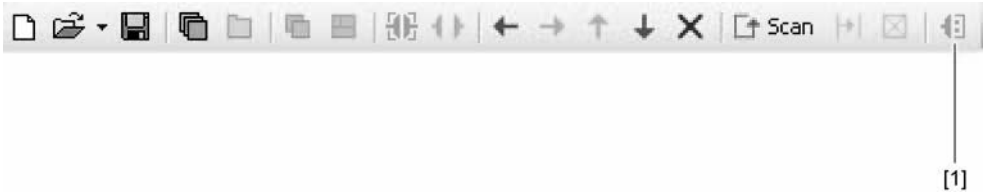
1. Connect the unit to a free USB port on your PC.
   Your PC will detect the new hardware and launch the hardware wizard.
2. Follow the instructions of the hardware wizard.
3. Click on [Browse] and go to the MOVITOOLS® MotionStudio installation folder.
4. Enter the following path:
   "..\Program Files\SEW\MotionStudo\Driver\SEW_USBWIN32_051120"
5. Click the [Next] button 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:

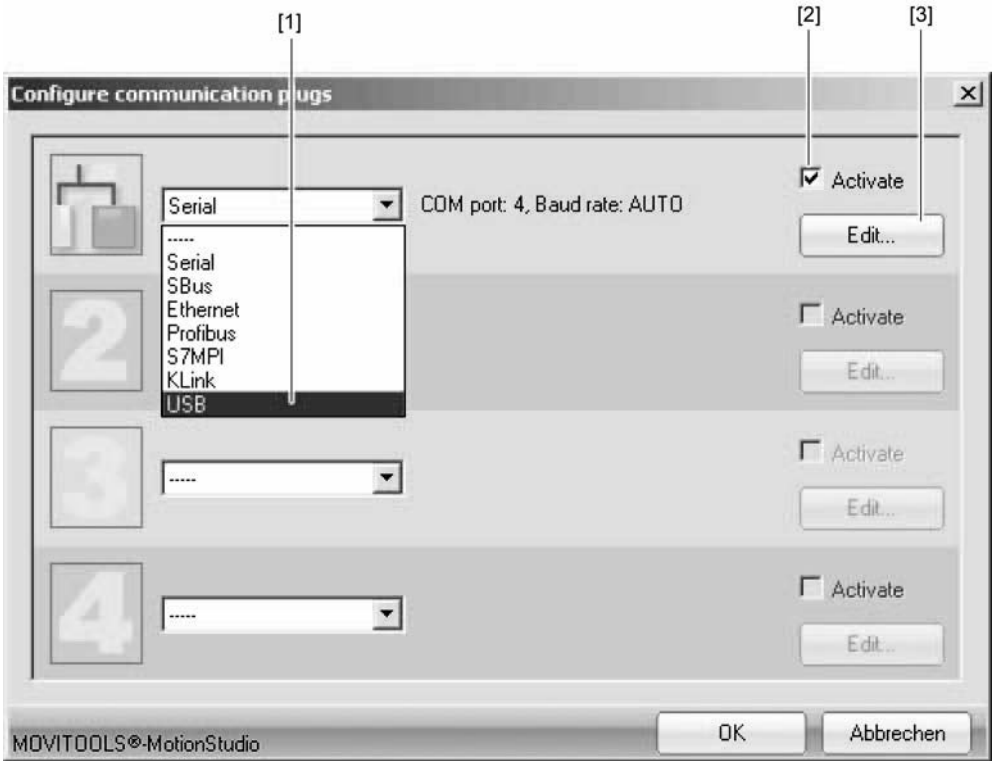
1. Click on "Configure communication plugs" [1] in the toolbar.



64341AXX

[1] "Configure communication plugs" icon

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



64743AEN

[1] "Communication type" selection field

[2] "Activated" check box

[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.

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:

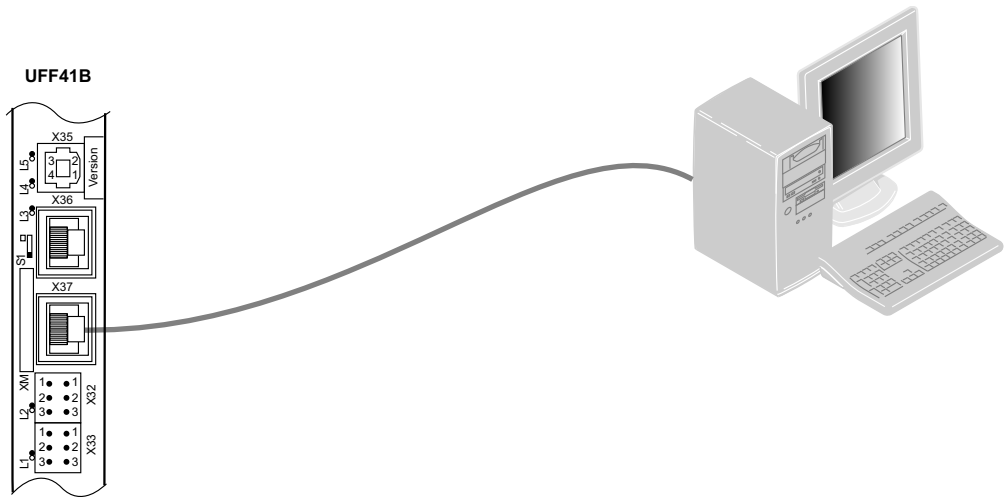
Communication parameters	Description	Note
Timeout	Waiting time in milliseconds that the master waits for a response from a slave after it has sent a request.	Default setting: 350 ms

12.5 Communication via Ethernet

12.5.1 Connecting the unit with the PC via Ethernet

Connecting the Ethernet interface of DHx41B/UFx41B to the PC

The following figure shows the connection between the PC/laptop and the DHx41B/UFx41B.



64420AXX

UFx41B/DHx41B can be connected either directly to the PC or via an Ethernet network.

The Ethernet interface X37 supports auto crossing and auto negotiation for baud rate and duplex mode. Set the IP parameters of UFx41B/DHx41B as described in chapter 4.5.

Adjust 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 check box 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.

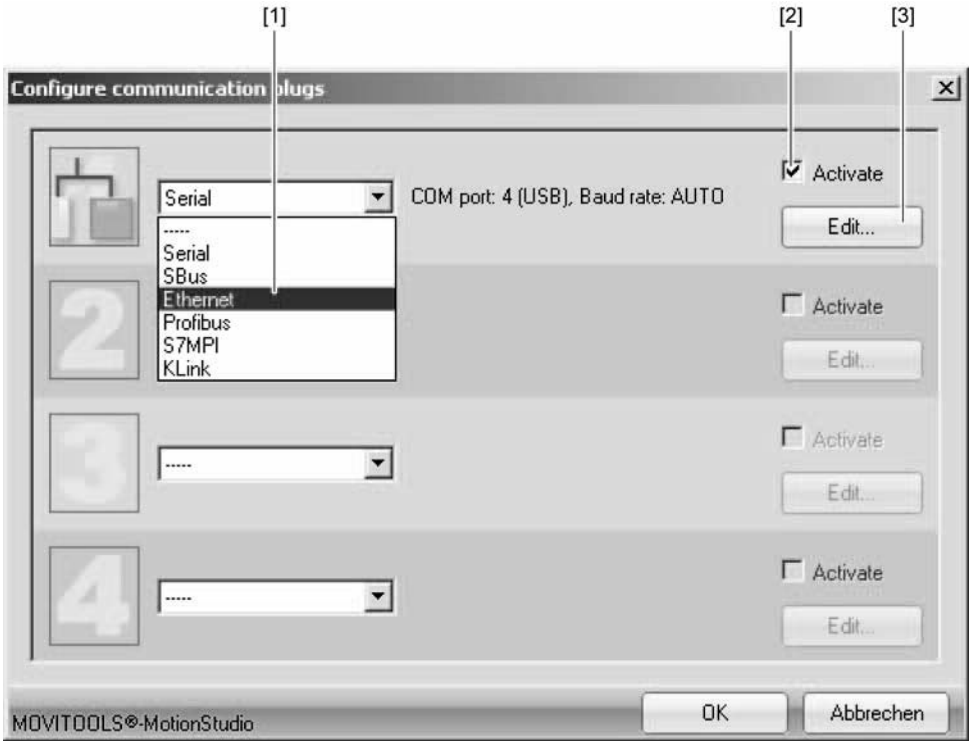
12.5.2 Configuring the communication channel via Ethernet

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
	SMLP stands for **S**imple **M**OVI**L**INK® **P**rotocol. It is the unit protocol from SEW-EURODRIVE.

12.5.3 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:

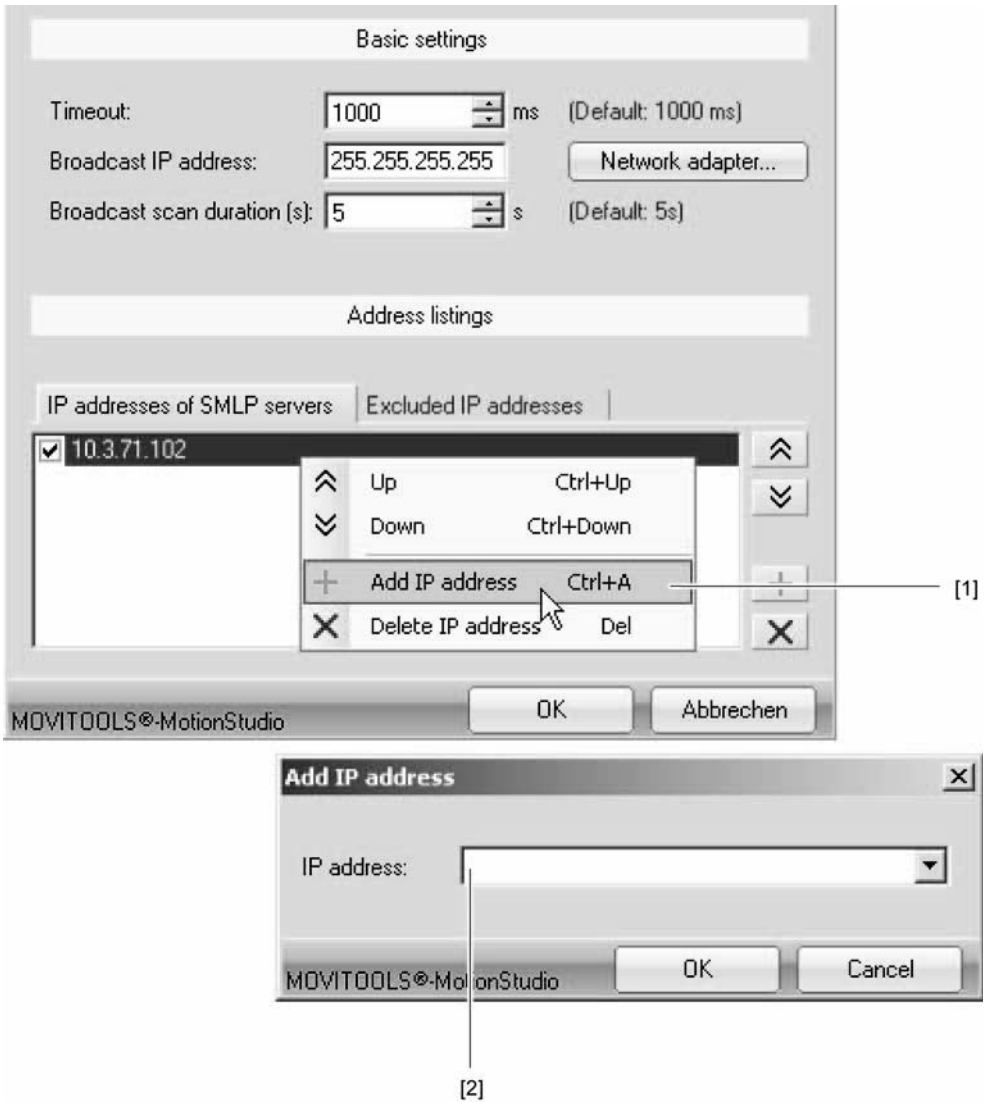
1. If necessary, change the preset communication parameters. Refer to the detailed description of the communication parameters for SMLP.

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].



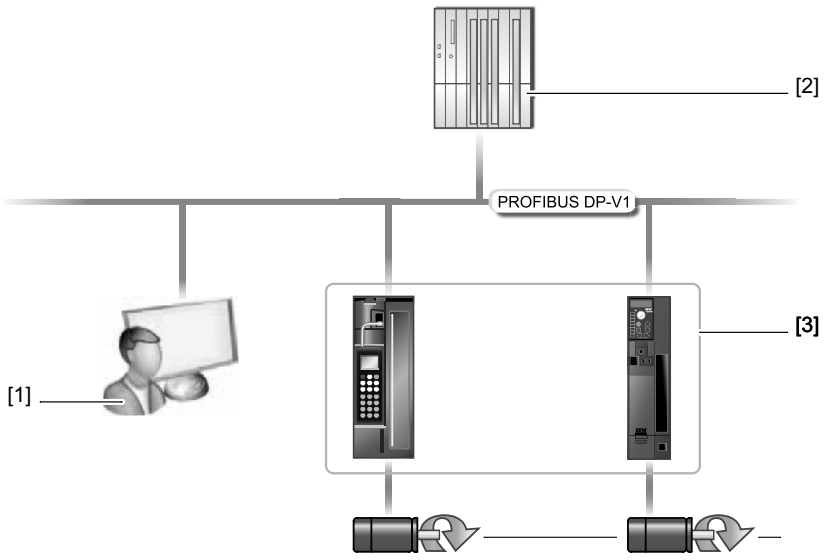
64352AEN

3. Add the IP address [2]

12.6 Communication via PROFIBUS DP/DP-V1

12.6.1 Communication via C2 master

Overview The figure shows the network with a direct PROFIBUS communication via C2 master:



[1] C2 master (as PC with installed softnet DP driver and installed PROFIBUS master card)

[2] C1 master

[3] Units (examples) with DP-V1 capable PROFIBUS interfaces

C2 master C2 master [1] can be a PC, for example, which you can use as diagnostic and visualization PC.

For this purpose, the PC has to be equipped wtih additional hardware and software as described in the next section.

Function The C2 master [1] sends parameter requests from MOVITOOLS® MotionStudio to the PROFIBUS interfaces in the units [3] via PROFIBUS (acyclic C2 services). In this case, SIMATIC S7 [2] does not perform any routing.

Advantage The C2 master works independently of the C1 master. This means you can establish a communication with your units even when the C1 master fails.

12.6.2 Additionally required hardware and software

Prerequisite

	TIP
	If you run and configure PROFIBUS stations in your network, you need additional hardware and software from Siemens. • Note the prerequisites regarding license rights for Siemens software products used. • Observe the documentation provided by Siemens for the hardware and software products used.

Required hardware

The following table shows the PROFIBUS master cards available from Siemens:

Designation of the PROFIBUS master card	Order number	Type of PROFIBUS master card
SIMATIC NET CP5611	6GK1561-1AA00	PCI card for PCs
SIMATIC NET CP5512	6GK1561-2AA00	PCMCIA card (32-bit card bus) for notebooks

Required software

The following table shows the software available from Siemens:

Designation of the software	Order number	Type of software
SIMATIC NET PB Softnet-DP Edition 2007	6GK1704-5DW00-3AE1	Driver package

Starting up hardware and software

Do the following to install the additionally required hardware and software:

1. Observe the documentation provided by Siemens for the hardware and software products used.
2. Install the PROFIBUS master card.
3. Install the software.

12.6.3 Parameterize C2 master with SIMATIC NET

SIMATIC NET versions and operating system

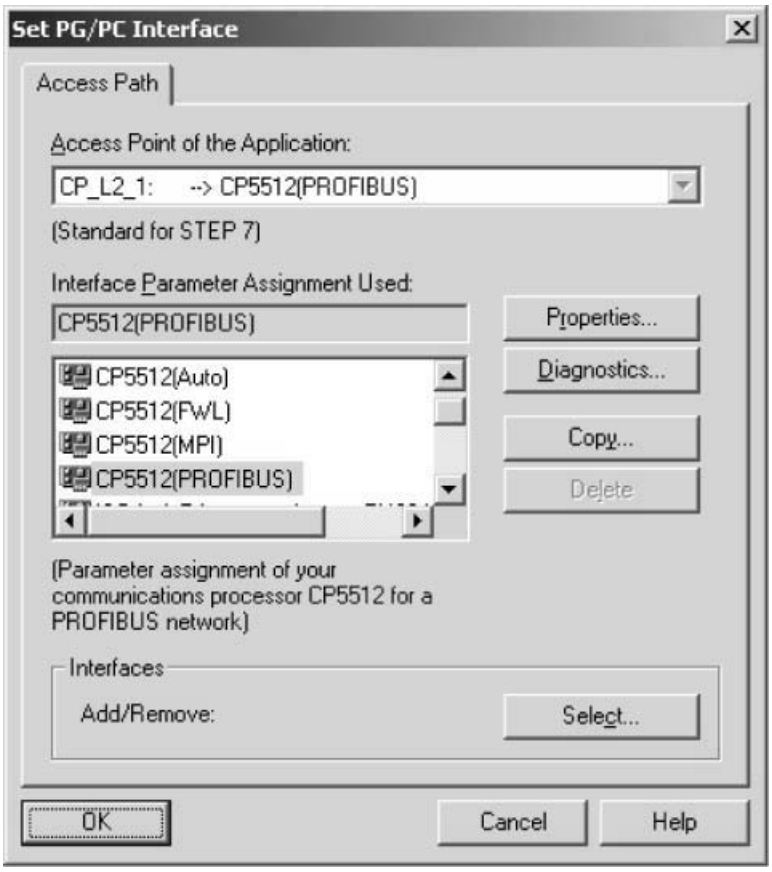
TIP
The following description might deviate slightly (in part due to the language) depending on the SIMATIC NET version and the operating system in use. This concerns the display and designations in windows as well as designations in the menu path of the start menu.

Starting SIMATIC NET and setting the PG/PC interface

Do the following to start SIMATIC NET:

1. From the Start menu of Windows, start the program "Set PG/PC Interface" under the following menu item:

   The "Set PG/PC interface" window opens:



12098AEN

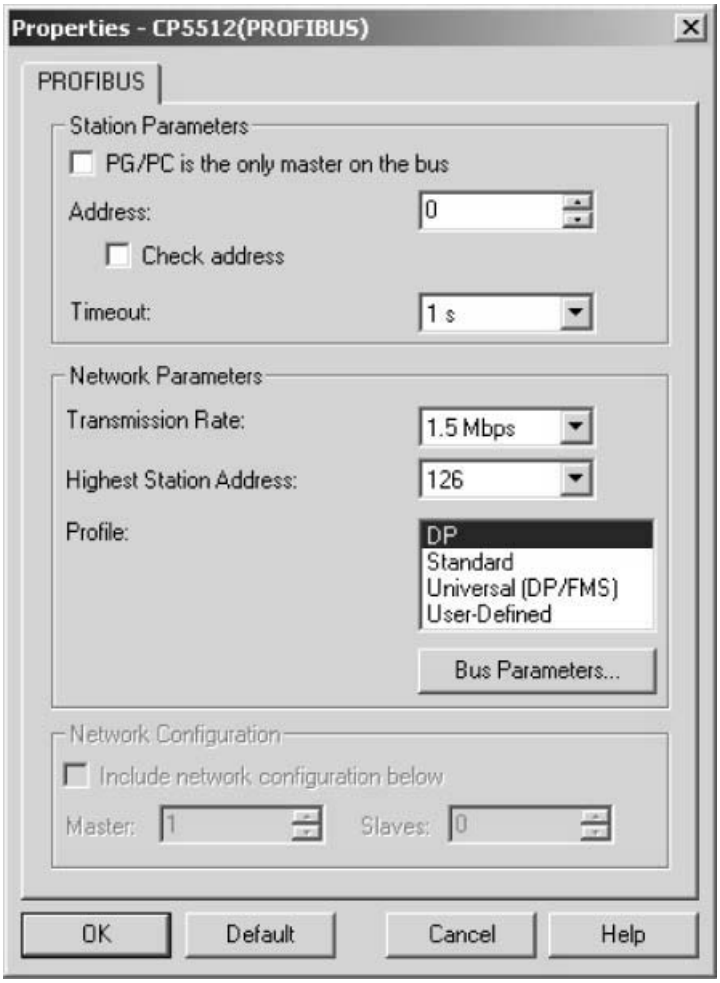
2. Set the access path of the application as shown in the figure.

TIP
If you cannot set the access path because the "Access point of the application" window is disabled, then the reason is the following: You have opened the "Set PG/PC interface" program from SIMATIC STEP 7 and have therefore occupied the access path. • Start the "Set PG/PC Interface" program from the Windows Start menu.

Parameterizing a C2 master

Proceed as follows to configure a C2 master:

1. In the "Set PG/PC interface" window, click on the [Properties] button. This opens the "Properties" window.



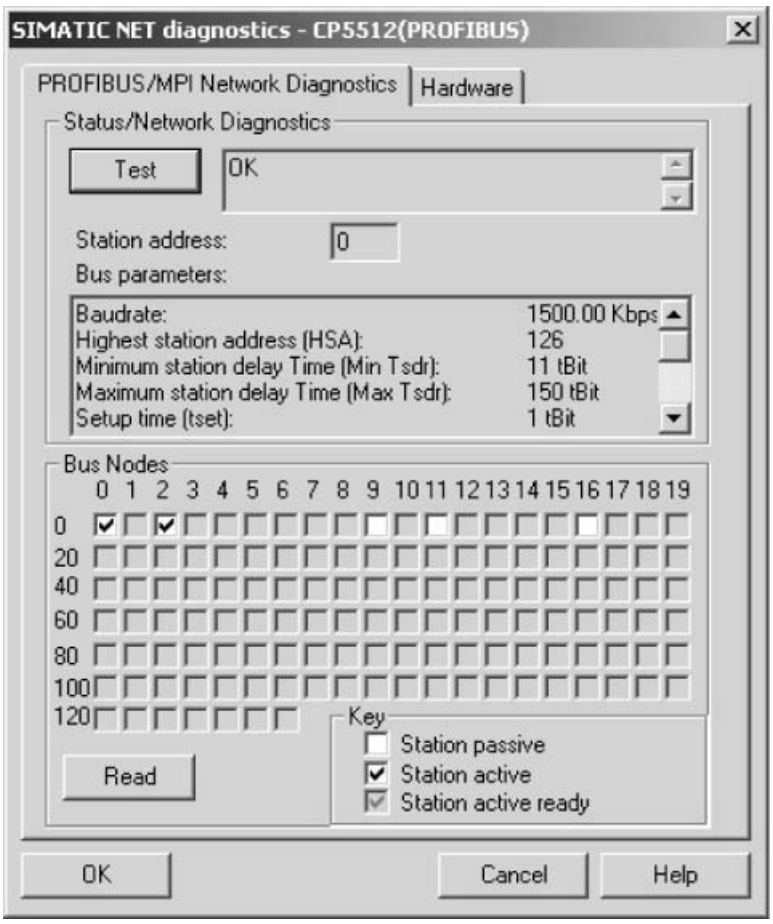
12097AEN

2. If a C1 master is active, disable the "PG/PC is the only master on the bus" check box.
3. Assign the PC a free address that is not yet reserved by other stations (masters or slaves).
4. Set the baud rate (transmission speed) matching your PROFIBUS network. If you operate a C1 master, set the baud rate of the C1 master.
5. Select "DP" as the profile or set the bus timing according to the existing PROFIBUS network.

Check the parameters of the PROFIBUS stations

Do the following to check the parameters of the PROFIBUS stations:

1. Close the "Properties" window to return to the "Set PG/PC interface" window.
2. Click the [Diagnostics] button.
   This opens the "SIMATIC NET diagnostics" window.



12096AEN

3. Check the parameters you have set. To do so, click on [Test].
   If your parameter setting is valid, "OK" will be displayed.
4. To have all bus stations displayed, click on [Read].
5. Make sure that all bus stations were parameterized correctly.
6. Open the MOVITOOLS® MotionStudio engineering software.
7. Set the communication parameters in MOVITOOLS® MotionStudio. Refer to the next section "Configuring communication via PROFIBUS".

12.6.4 Configuring communication via PROFIBUS

Prerequisites

	TIP
	The following steps describe only how you configure PROFIBUS communication in MOVITOOLS® MotionStudio. • **First** make all the required settings in the project planning software. Refer to the previous section "Configuring C2 master with SIMATIC NET".

Configuring a communication channel via PROFIBUS

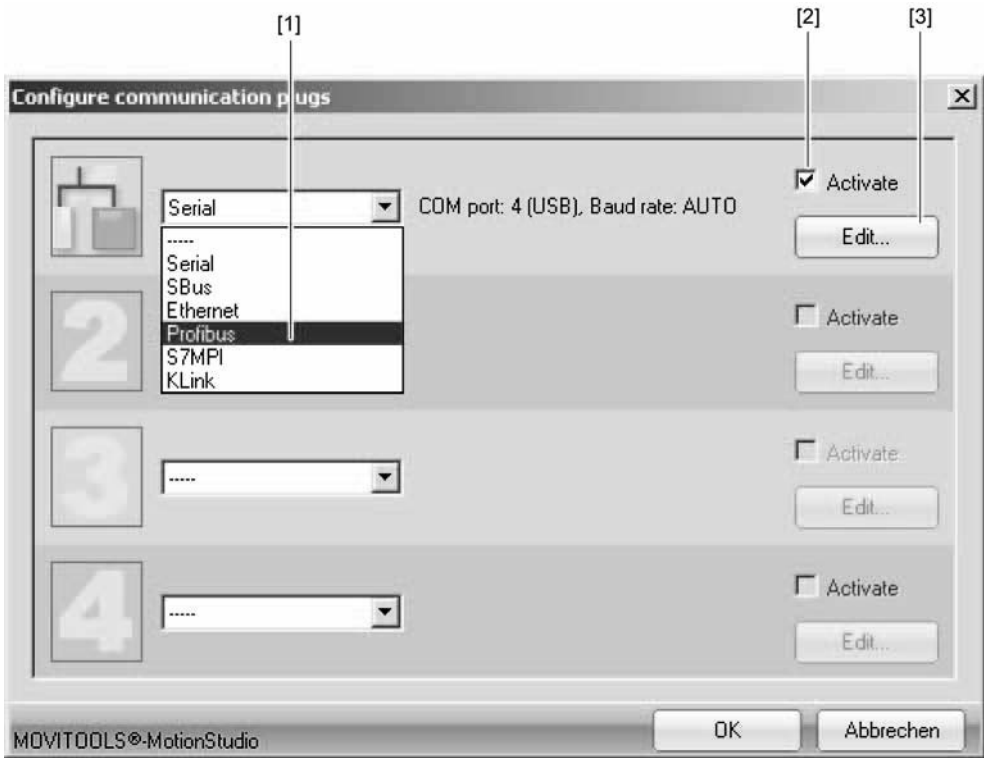
Proceed as follows to configure PROFIBUS communication:

1. Make sure that all the required settings have been made in the project planning software.
2. Start MOVITOOLS® MotionStudio and create a project following the instructions described in the section "First Steps".
3. Click on "Configure communication plugs" [1] in the toolbar.

[1] "Configure communication plugs" icon

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

4. From the list [1], select "PROFIBUS" as the communication type.



64619AEN

[1] "Communication type" selection list

[2] "Activated" check box

[3] [Edit] button

In the example, "PROFIBUS" is activated as the communication type for the first communication channel [2].

5. Click [Edit] [3] in the right section of the window.

12095AEN

6. Select the "Start automatically" control field if you want to launch the PROFIBUS server every time the SEW Communication Server is started.
7. Click the [Restart server] button to start the PROFIBUS server.

   Windows displays the activated PROFIBUS server using the following ICON in the status bar:

12.6.5 Communication parameters for PROFIBUS DP/DP-V1

The following table describes the communication parameters for the PROFIBUS DP/DP-V1 communication channel:

Communication parameters	Description	Note
PROFIBUS server	Select the "Start automatically" control field if you want to launch the PROFIBUS server every time the SEW Communication Server is started.	The Windows status bar displays the active PROFIBUS server

12.7 Executing functions with the units

12.7.1 Parameterizing units in the parameter tree

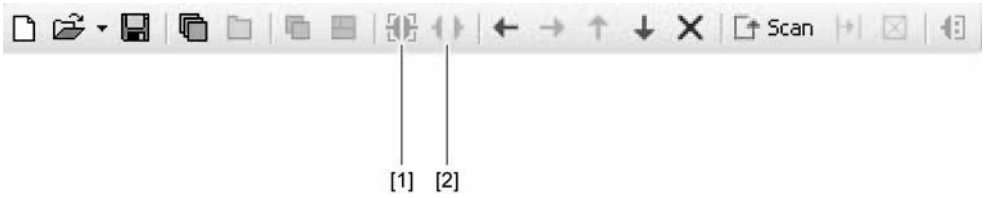
The parameter tree displays all unit parameters arranged in folders.

You can manage the unit parameters using the context menu or toolbar. The following chapter describes how to read or change unit parameters.

12.7.2 Reading/changing unit parameters

To read or change unit parameters, proceed as follows:

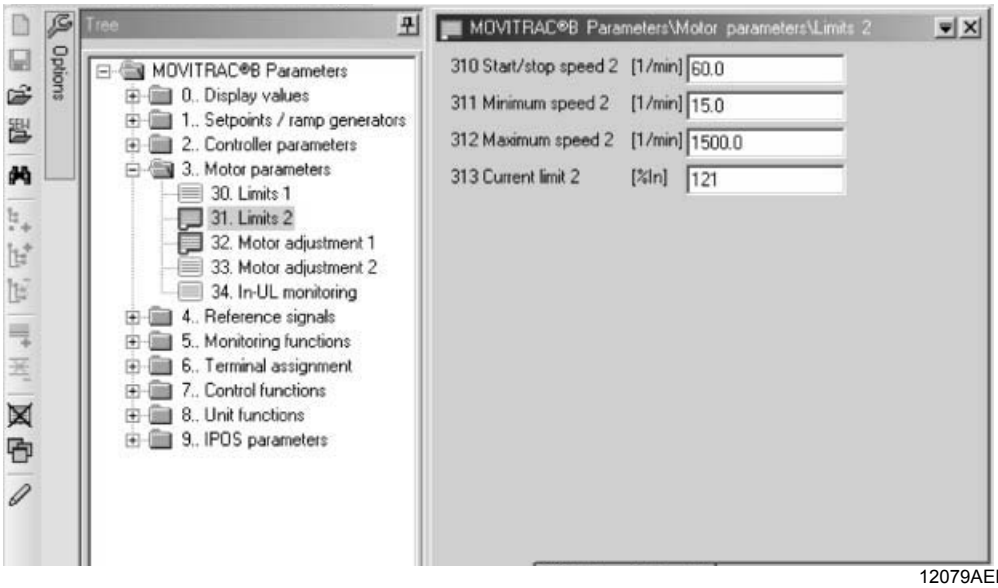
1. Switch to the required view (project view or network view).
2. Select the communication mode:
   - Click the [Switch to online mode] button [1] if you want to read or change parameters directly on the **unit**.
   - Click the [Switch to offline mode] button [2] if you want to read or change parameters 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.



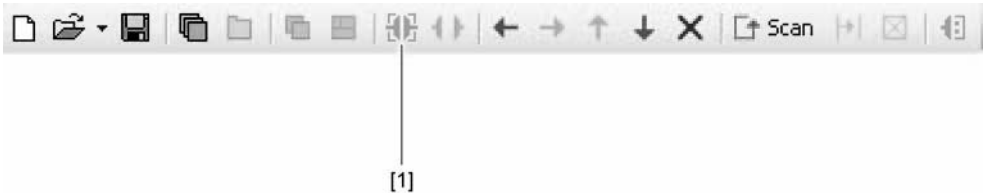
12079AEN

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.

12.7.3 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
	• 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.

13 Error Diagnostics for Operation on PROFIBUS DP-V1

13.1 Diagnostic procedure for PROFIBUS-DP

Diagnostic problem: MOVI-PLC® advanced DHF41B is not working on the PROFIBUS.
Initial status:
- The MOVI-PLC® *advanced* DHF41B control card is connected physically to PROFIBUS.
- The MOVI-PLC® *advanced* DHF41B control card was configured in the DP master and bus communication is active.

↓

Is the bus connector plugged in? — No → **[A]**

Yes ↓

Response of the LED PROFIBUS fault? — OFF → **[B]**; ON → **[C]**

FLASHES ↓

MOVI-PLC® *advanced* DHF41B recognizes the baud rate but was either not configured in the PROFIBUS DP master or was configured incorrectly.

↓

Check whether the configured PROFIBUS address is the same as the address set on the DIP switches.

↓

PROFIBUS addresses identical? — No → **[D]**

Yes ↓

You might have configured an incorrect unit type or defined the configuration incorrectly.

↓

Delete the configuration for MOVI-PLC® *advanced* DHF41B from the DP network.

↓

Configure MOVI-PLC® *advanced* DHF41B again choosing the unit designation "MOVI-PLC".
Use a predefined configuration to simplify the process. Do not change any of the preset configuration data.
Assign the address range for your control system.

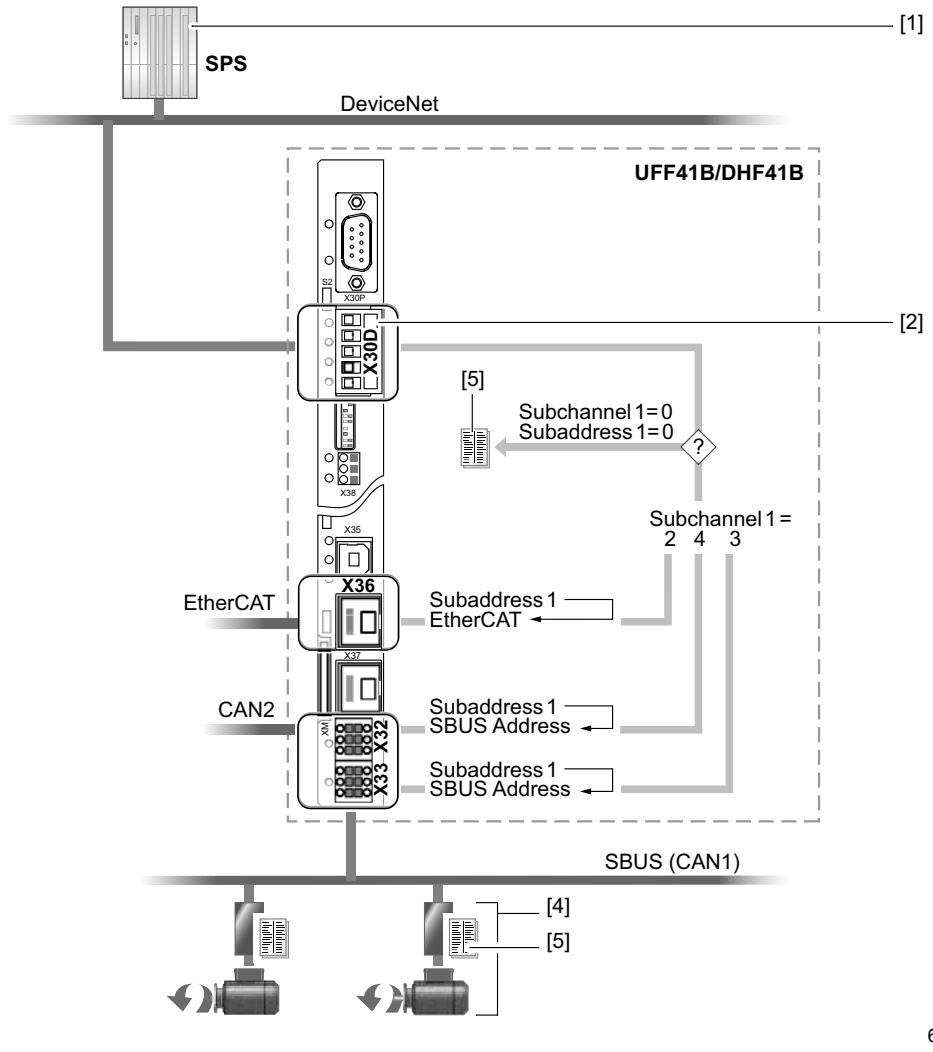
↓

Load the configuration into the PROFIBUS DP master and start the bus communication again.

[A]	Check the bus cabling.
[B]	MOVI-PLC® *advanced* DHF41B is currently exchanging data cyclically with the PROFIBUS DP master.
[C]	TMOVI-PLC® *advanced* DHF41B does **not** recognize the baud rate! Check the bus cabling.
[D]	Adapt the bus addresses.

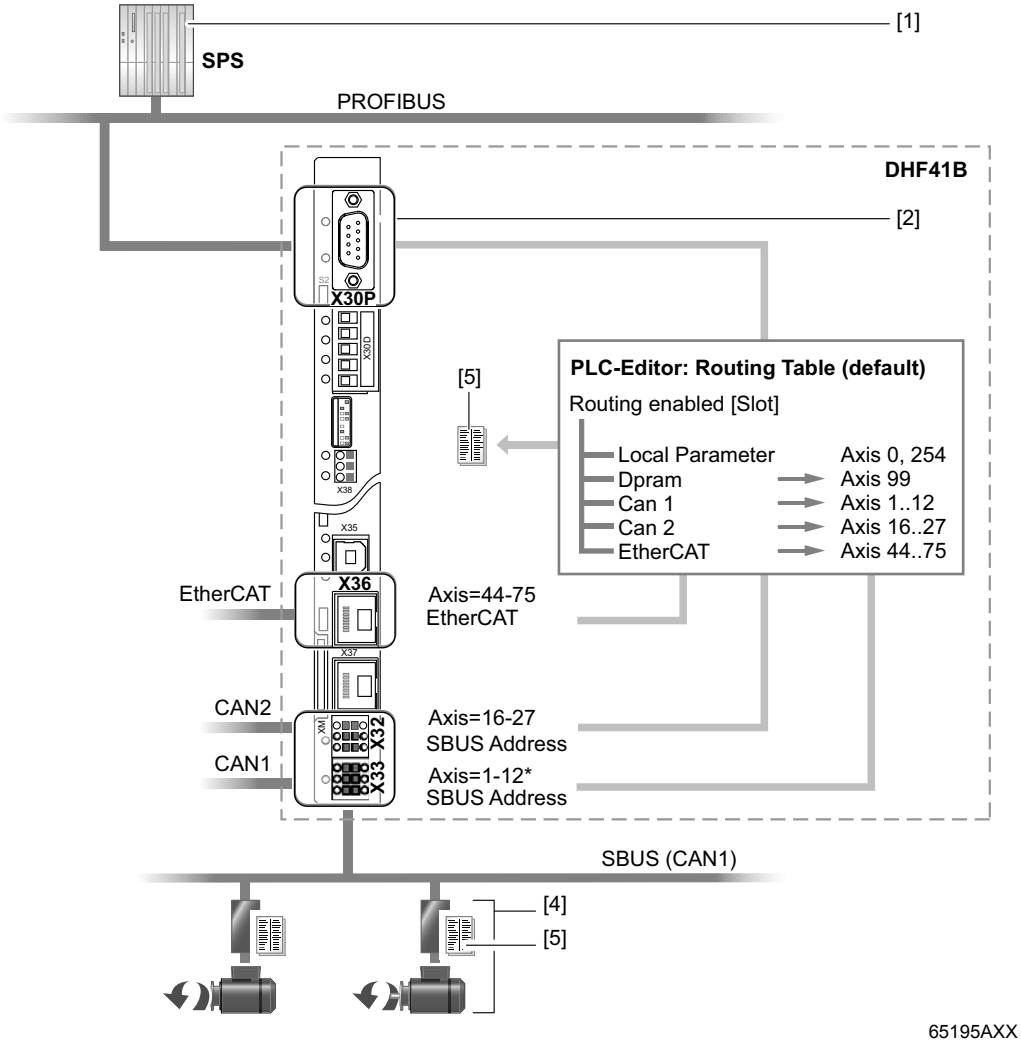
14 Appendix

14.1 Parameter access to lower-level units via DeviceNet



[1] PLC with DeviceNet scanner (master)
[2] DeviceNet interface
[4] SEW inverter with SBus interface
[5] Index and parameter list of the unit

14.2 Parameter access to lower-level units via PROFIBUS DP-V1



* SBus address 15 must not be used when engineering via PROFIBUS or parameter services via PROFIBUS.

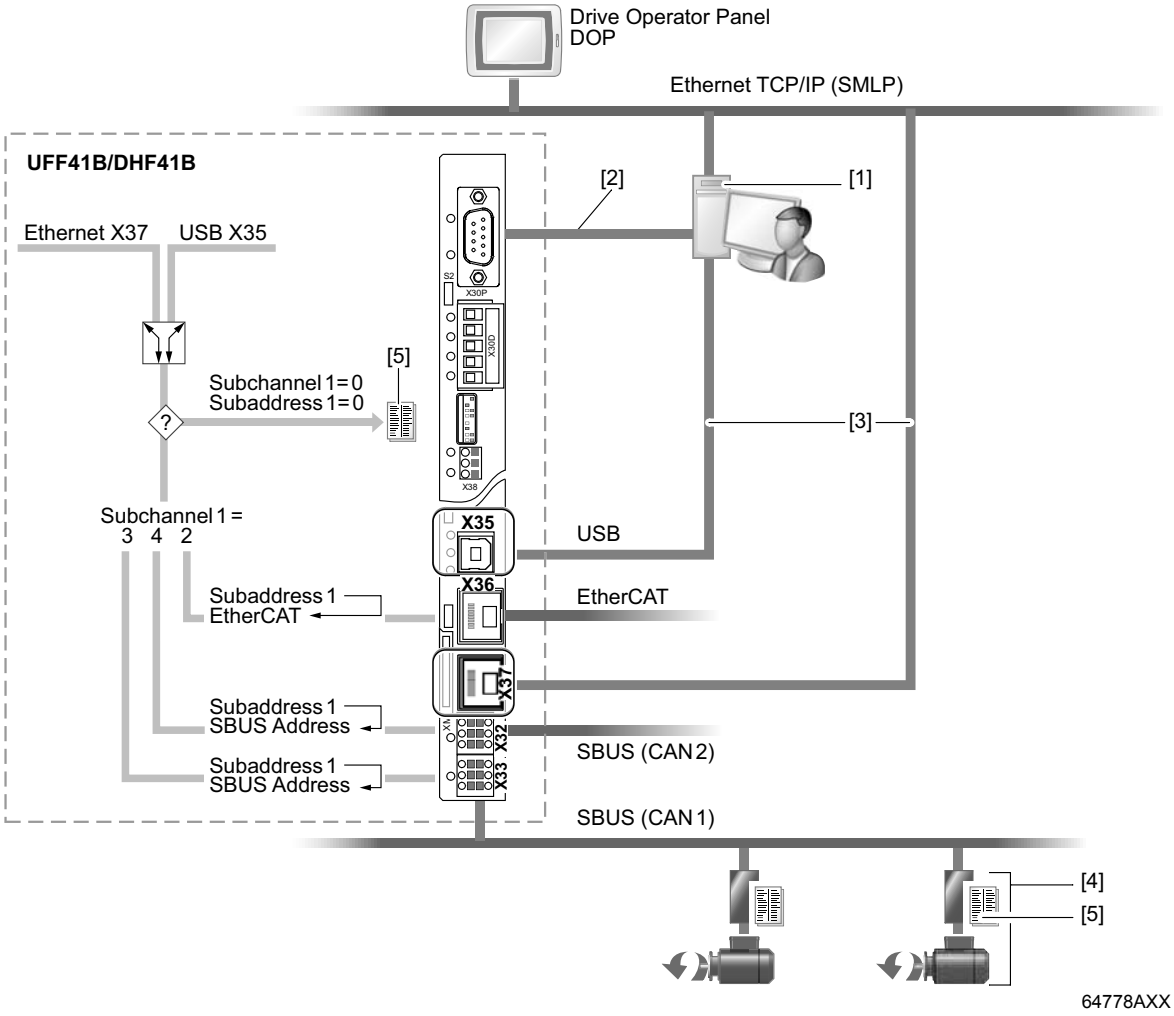
[1] SPS with PROFIBUS DP-V1 master

[2] PROFIBUS interface

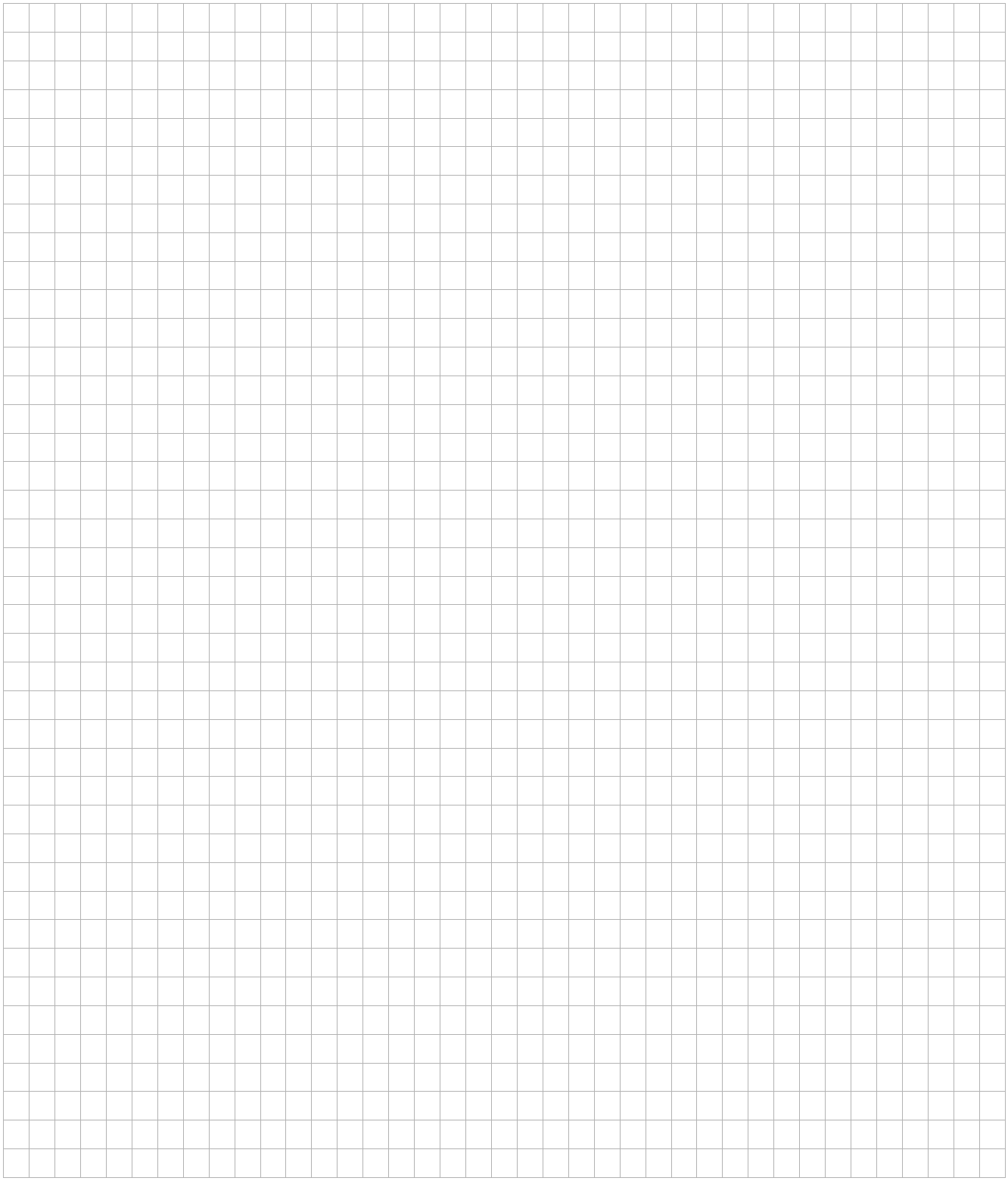
[4] SEW inverter with SBus interface

[5] Index and parameter list of the unit

14.3 Parameter access to lower-level units via engineering interfaces



[1] Engineering PC
[2] PROFIBUS interface (for engineering)
[3] USB/Ethernet engineering interface
[4] SEW inverter with SBus interface
[5] Index and parameter list of the unit



How we're driving the world

With people who think fast and develop the future with you.

With a worldwide service network that is always close at hand.

With drives and controls that automatically improve your productivity.

With comprehensive knowledge in virtually every branch of industry today.

With uncompromising quality that reduces the cost and complexity of daily operations.

SEW-EURODRIVE
Driving the world

With a global presence that offers responsive and reliable solutions. Anywhere.

With innovative technology that solves tomorrow's problems today.

With online information and software updates, via the Internet, available around the clock.

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023 · D-76642 Bruchsal / Germany
Phone +49 7251 75-0 · Fax +49 7251 75-1970
sew@sew-eurodrive.com

→ **www.sew-eurodrive.com**