SEW EURODRIVE

Manual

MOVIKIT® Power and Energy Solutions

PowerMode

Edition 03/2020

26624737/EN

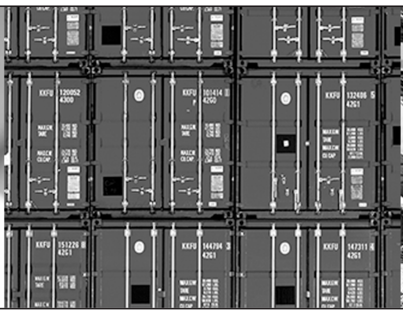


Table of contents

26624737/EN – 03/2020

26624737/EN – 03/2020

1 General information

1.1 About this documentation

This documentation is an integral part of the product. The documentation is intended for all employees who perform work on the product.

Make sure this documentation is accessible and legible. Ensure that persons responsible for the systems and their operation as well as persons who work with the product independently have read through the documentation carefully and understood it. If you are unclear about any of the information in this documentation, or if you require further information, contact SEW-EURODRIVE.

1.2 Content of the documentation

The descriptions in this documentation apply to the software and firmware versions applicable at the time of publication. These descriptions might differ if you install later software or firmware versions. In this case, contact SEW-EURODRIVE.

1.3 Structure of the safety notes

1.3.1 Meaning of signal words

The following table shows the grading and meaning of the signal words for safety notes.

Signal word	Meaning	Consequences if disregarded
⚠ **DANGER**	Imminent hazard	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 product or its environment
INFORMATION	Useful information or tip: Simplifies handling of the product.	

1.3.2 Structure of section-related safety notes

Section-related safety notes do not apply to a specific action but to several actions pertaining to one subject. The hazard symbols used either indicate a general hazard or a specific hazard.

This is the formal structure of a safety note for a specific section:

SIGNAL WORD

Type and source of hazard.

Possible consequence(s) if disregarded.

- Measure(s) to prevent the hazard.

26624737/EN – 03/2020

Meaning of the hazard symbols

The hazard symbols in the safety notes have the following meaning:

Hazard symbol	Meaning
	General hazard
	Warning of dangerous electrical voltage

1.3.3 Structure of embedded safety notes

Embedded safety notes are directly integrated into the instructions just before the description of the dangerous action.

This is the formal structure of an embedded safety note:

⚠ **SIGNAL WORD!** Type and source of hazard. Possible consequence(s) if disregarded. Measure(s) to prevent the hazard.

1.4 Decimal separator in numerical values

In this document, a period is used to indicate the decimal separator.

Example: 30.5 kg

1.5 Rights to claim under limited warranty

Read the information in this documentation. This is essential for fault-free operation and fulfillment of any rights to claim under limited warranty. Read the documentation before you start working with the product.

1.6 Product names and trademarks

The brands and product names in this documentation are trademarks or registered trademarks of their respective titleholders.

1.6.1 Trademark of Beckhoff Automation GmbH

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

1.7 Copyright notice

1.8 Other applicable documentation

Observe the corresponding documentation for all further components.

Always use the latest edition of the documentation and the software.

The SEW-EURODRIVE website (www.sew-eurodrive.com) provides a wide selection of documents for download in various languages. If required, you can also order printed and bound copies of the documentation from SEW-EURODRIVE.

1.9 Short designation

The following short designations are used in this documentation:

Type designation	Short designation
MOVIKIT® Power and Energy Solutions PowerMode	MOVIKIT® PowerMode
MOVIKIT® Power and Energy Solutions PowerMode	Software module
MDP92A power supply module with controlled DC link voltage	MDP92A
DC/DC converter module MDE90A	MDE90A

2 Safety notes

2.1 Preliminary information

The following general safety notes serve the purpose of preventing injury to persons and damage to property. They primarily apply to the use of products described in this documentation. If you use additional components, also observe the relevant warning and safety notes.

2.2 Target group

Software specialist

Any work with the software may only be performed by a specialist with suitable training. A specialist in this context is someone who has the following qualifications:

- Appropriate training
- Knowledge of this documentation and other applicable documentation
- SEW-EURODRIVE recommends additional training for products that are operated using this software.

2.3 Network security and access protection

A bus system makes it possible to adapt electronic drive technology components to the particulars of the machinery within wide limits. There is a risk that a change of parameters that cannot be detected externally may result in unexpected but not uncontrolled system behavior and may have a negative impact on operational safety, system availability, or data security.

Ensure that unauthorized access is prevented, especially with respect to Ethernet-based networked systems and engineering interfaces.

Use IT-specific safety standards to increase access protection to the ports. For a port overview, refer to the respective technical data of the device in use.

2.4 Designated use

The MOVIKIT® Power and Energy Solutions PowerMode is used in conjunction with the hardware components of the Power and Energy Solutions product range to provide intelligent power and energy management.

Use the device-independent MOVISUITE® engineering software to start up and configure the devices and to download the complete configuration to a MOVI-C® CONTROLLER.

Observe the documentation for the components used.

Unintended or improper use of the product may result in severe injury to persons and damage to property.

26624737/EN – 03/2020

3 System description

3.1 Module description

Introduction

The Power and Energy Solutions product range expands SEW-EURODRIVE's current line of MOVIDRIVE® modular inverters to include intelligent power and energy management components. In addition to hardware components, the product range includes the MOVIKIT® modular system with software modules for optimum integration at the control software level.

MOVIKIT® PowerMode

The MOVIKIT® PowerMode is designed for applications with energy storage units and provides the user with the following functions, among others:

- Operating the MDP92A and MDE90A devices from the MOVI-C® CONTROLLER
- Setting a state of charge in the DC link
- Disconnection and synchronization of disconnectable energy storage units
- Discharging an energy storage unit via a connected braking resistor
- Connection of diagnostic interfaces of the energy storage units
- Reading measured values from the devices for use in the IEC program
- Monitoring of the multi-stage fuse concept

INFORMATION

The automatic code generation automatically integrates all blocks required for operation.

3.2 Functions

Overview of functions:

- **Recording of power and energy data**
  - Communication with Power and Energy Solutions devices (MDP92A, MDE90A) for detecting power and energy data at the connection point to the power grid and to the energy storage unit
  - Communication with MOVIDRIVE® modular for determining the power demand of the connected application
  - Determining the total power and energy balance including AC loads
- **Managing the DC link**
  - Specifying voltage setpoints of the DC link during operation
  - Configuring a maximum power output to the DC link during operation
  - Calculating the power of the DC link capacity
  - Use as a central node for supply and output power
  - Discharge of the DC link capacity
- **Managing the AC connection**
  - Limiting the grid power (peak shaving)
  - Measuring phase-to-phase voltages
  - Calculating the current consumption of the individual line phases (applies only to symmetrical three-phase supply systems)

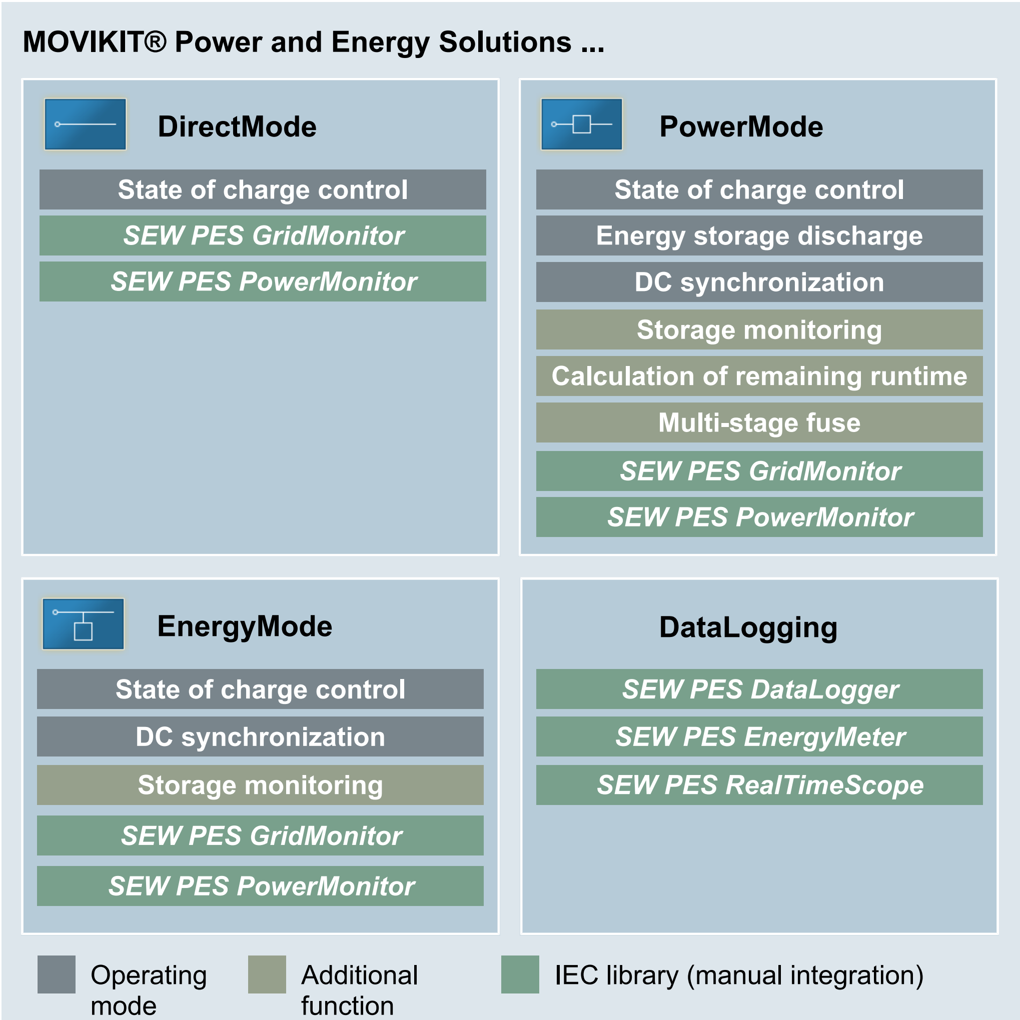
26624737/EN – 03/2020

  – Phase and power failure detection and calculation of the remaining system runtime
- **Managing the energy storage unit**
  – Providing storage-relevant data (temperature, voltage, overvoltage detection)
  – Evaluating diagnostic interfaces of selected energy storage units available on the market
  – Aggregation of data when using storage units consisting of several modules
  – Transferring the storage state to the application program for further processing
  – Discharging energy storage units
  – Synchronization, connection and disconnection of disconnectable energy storage units
  – Monitoring and control of multi-stage fuse concepts for specific energy storage types or sizes

26624737/EN – 03/2020

3.3 Components

The following figure provides an overview of the software modules of the MOVIKIT® Power and Energy Solutions modular system:



9007230506411147

3.3.1 Operating modes

MOVIKIT® PowerMode provides the following operating modes:

- **State of charge control (StateOfChargeControl)**

  Operating mode for regulating a specific state of charge in the DC link or energy storage unit by means of a connected device.

- **Energy storage discharge (StorageDischarger)**

  Operating mode for discharging the DC link via a braking resistor connected to the MDP92A power supply module.

- **DC synchronization**

  Operating mode for synchronizing and connecting or controlled disconnection of separable energy storage units of the MDP92A and MDE90A devices.

For further information, refer to chapter "Operating modes" (→ 🗎 31).

3.3.2 Additional functions

MOVIKIT® PowerMode offers the following additional functions with the automatic code generation function in the main function block:

- **StorageMonitor**

  Additional function for connecting diagnostic interfaces of commercially available storage units

- **RemainingRuntimeCalculator**

  Additional function for calculating the remaining system runtime after power failure

- **MultipleFuseControl**

  Additional function for monitoring the current voltage and the status of the energy storage unit. If necessary, the corresponding fuse is switched on or off.

For further information, refer to chapter "Additional functions" (→ 🗎 46).

3.3.3 IEC libraries

MOVIKIT® PowerMode also provides the following IEC libraries for manual integration:

- **SEW PES GridMonitor**

  Function blocks for detecting AC loads via measuring terminals

- **SEW PES PowerMonitor**

  Central power node for balancing DC link power flows.

For further information, refer to chapter "IEC libraries" (→ 🗎 55).

4 Project planning information

4.1 Requirement

Correct project planning and proper installation of the devices are required for successful startup and operation.

For detailed project planning information, refer to the documentation of the respective devices.

4.2 Hardware

The following hardware is required:

- MOVI-C® CONTROLLER (recommended as of performance class UHX45A)
- MDP92A power supply module with controlled DC link voltage

  or

  DC/DC converter module MDE90A
- One or more MDC90A capacitor modules

  or

  Energy storage based on double-layer capacitor technology

  (if required with diagnostic interface for full scope of functions)

4.3 Software

The following software is required:

- MOVISUITE® engineering software

  (includes MOVIRUN® flexible)

For more detailed information on the hardware requirements of the individual software components, see the documentation for the respective software.

4.4 Licensing

The following licenses are available and are required:

- MOVIRUN® flexible

  License for the software platform MOVIRUN® flexible
- MOVIKIT® PowerMode – performance license

  License for software module MOVIKIT® PowerMode

  – SMK1402-040 (for UHX45A)

  – SMK1402-060 (for UHX65A)

  – SMK1402-080 (for UHX85A)

For further information on licensing, refer to the document "MOVI-C® Software Components". You can download the document from the SEW-EURODRIVE website (www.sew-eurodrive.com).

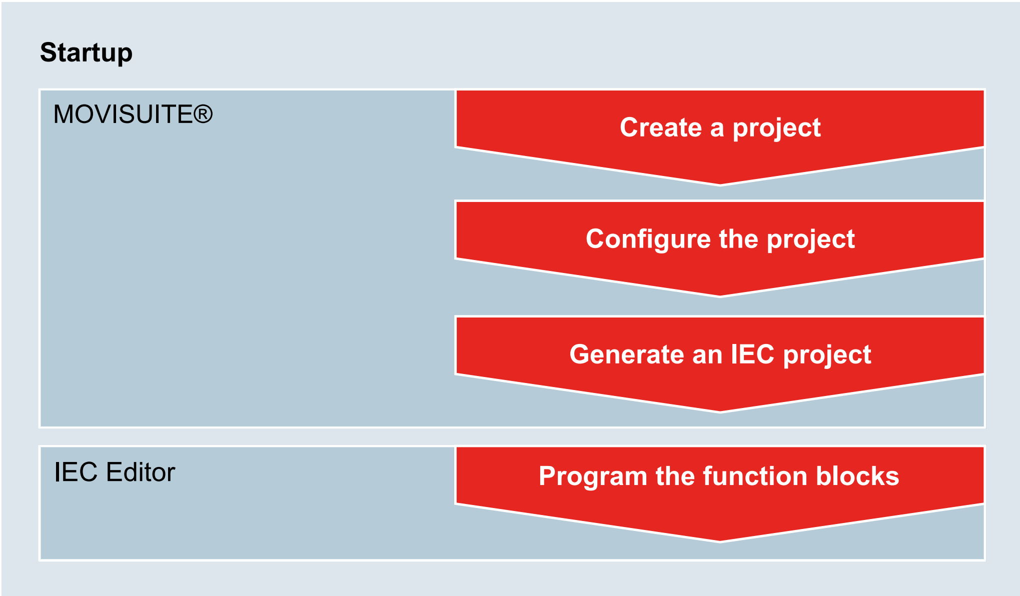
5 Startup

5.1 Requirements

- Check the installation of the inverters and, if installed, also check the encoder connection.
- Observe the installation notes in the documentation of the respective device and software components.
- The devices to be started up are displayed in MOVISUITE®.

5.2 Startup procedure

The schematic diagram below shows the startup procedure:



18014426855028235

The startup steps specific to these software modules are explained in detail in the following chapters of this manual. For startup, also observe the documentation of all the other components in use.

26624737/EN – 03/2020

5.3 Configuring a project

INFORMATION

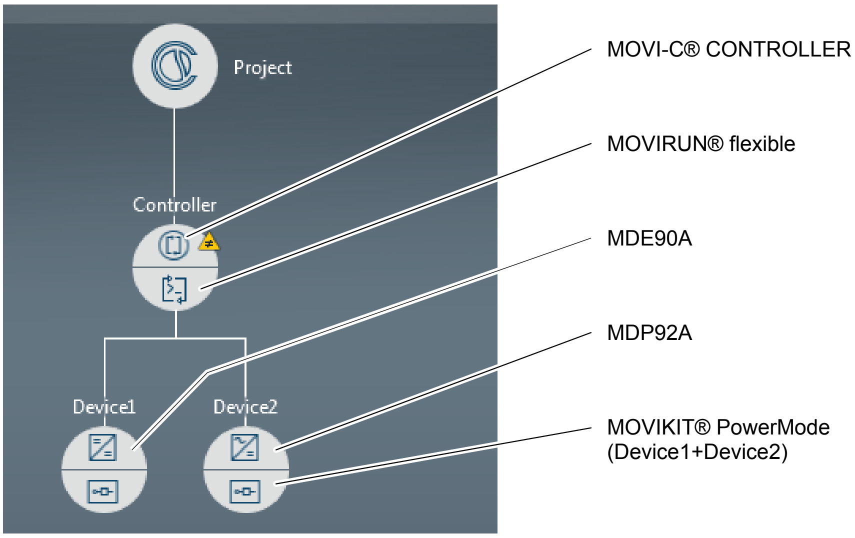
For detailed information on how to use the MOVISUITE® engineering software, refer to the corresponding documentation.

✓ A MOVISUITE® project has been created and is open.

1. Add required device nodes, software nodes (MOVI-C® SoftwareNode) and software modules to the project.
   ⇨ See "Example project".
2. Configure the added devices or software modules. If available, observe the specific notes in the following chapters that apply to MOVIKIT® Power and Energy Solutions PowerMode. For detailed information on the configuration of devices or other software modules, refer to the respective documentation.

5.3.1 Example project

The following figure shows an example project:



31155397643

26624737/EN – 03/2020

5.3.2 Adding MOVIKIT® Power and Energy Solutions PowerMode

INFORMATION

For detailed information on how to use the MOVISUITE® engineering software, refer to the corresponding documentation.

✓ A MOVISUITE® project has been created and is open.

1. Click on the empty software module section of the required node.
   ⇨ The catalog section opens and displays the available software modules.
2. In the catalog section, click on MOVIKIT® Power and Energy Solutions Power-Mode.
   ⇨ A context menu opens.
3. Select the version from the respective drop-down list in the context menu and confirm your selection with [Apply].

⇨ MOVIKIT® Power and Energy Solutions PowerMode is assigned to the node, the configuration is created, and the basic settings are made.

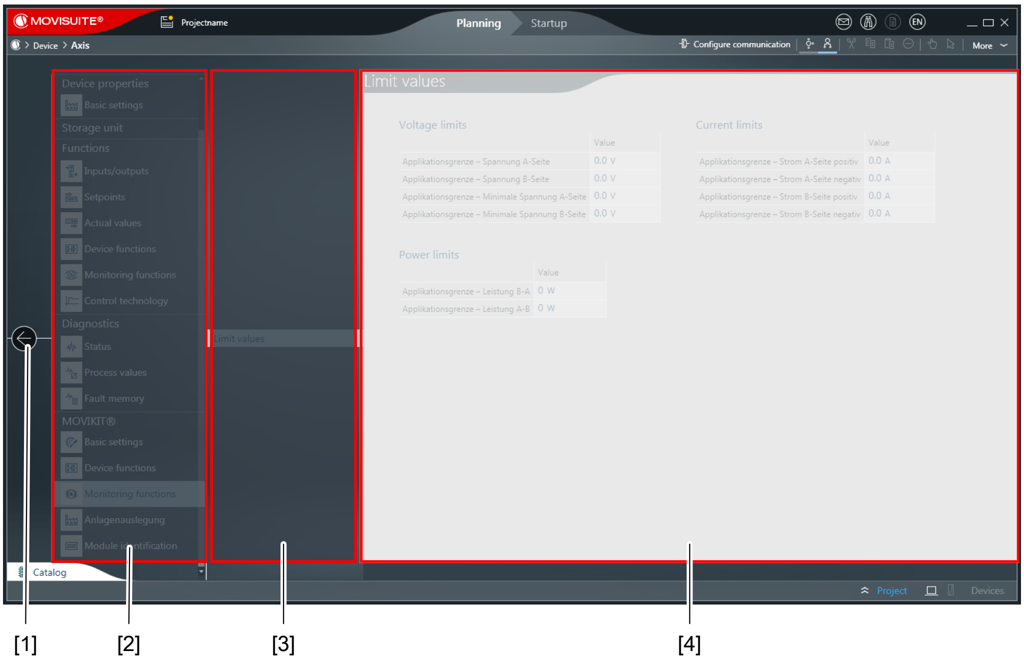
26624737/EN – 03/2020

5.3.3 Configuring MOVIKIT® Power and Energy Solutions PowerMode

INFORMATION

For detailed information on how to use the MOVISUITE® engineering software, refer to the corresponding documentation.

1. In MOVISUITE®, click on MOVIKIT® Power and Energy Solutions PowerMode.
   ⇨ The configuration menus of the software module are displayed. The configuration menus are explained in the following subchapters.



9007228165413771

[1] Button to return to the project overview
[2] Main menu of the software module configuration (MOVIKIT® section)
[3] Submenus of the configuration
[4] Setting fields of the respective submenu

2. Configure the software module using the respective setting fields.
3. Click button [1] after having completed the configuration.
   ⇨ The project overview is displayed.

INFORMATION

For the changes made to the configuration to take effect, you have to update the configuration data. To do so, click [Update configuration data] in the respective notification at the node or in the context menu of the MOVI-C® CONTROLLER. The MOVI-C® CONTROLLER is stopped and restarted to update the configuration data.

Basic settings

Parameter group	Description
Initialization	Initialization of the device for operation of the software module

Device functions

FCB 55 Voltage control

Parameter group	Value
Voltage control	Activation of boost mode **Default value: No**
Voltage-controlled output stage end	Selection of the controlled output stage end **Information:** Setting only available with the MDE90A. **Default value: No**

Monitoring functions

Limit values

Configuration of the application limits correspond to the application.

Parameter group	Description
Voltage limits **Information:** To control the boost mode via the software module, the minimum voltage of each side that uses the boost mode must be set to "0".	Voltage limits in [V] are set using the following parameters: • A-side and B-side voltage • Minimum A-side and B-side voltage **Default value: 0.0 V**
Current limits	Current limits in [A] are set using the following parameters: • Positive A-side and B-side current • Negative A-side and B-side current **Default value: 0.0 A**
Power limits	Power limits in [kW] are set using the following parameters: • B-A or A-B power **Default value: 0 kW**

26624737/EN – 03/2020

System design

INFORMATION

Generally, you will receive the values to be configured here during project planning.

Parameter group	Description
Boost charging power **INFORMATION:** To control the boost mode via the software module, the boost mode must be enabled in the corresponding configuration menu of the device.	The boost charging power is set using the following parameters: • Activation of boost mode **Default value: Off** • Setting of the application limit B-A in [W] **Default value: 0 W**
Discharging limit	The discharging limit is set using the following parameters: • Application limit – minimum voltage A-side in [V] **Default value: 0.0 V** • Minimum A-side voltage – source **Default value: Application limit – minimum A-side voltage**
Charging limit	The charging limit is set using the following parameters: • Application limit – voltage A-side in [V] **Default value: 0.0 V** • Maximum A-side voltage – source **Default value: Application limit – A-side voltage**
Current limits	Current limits are set using the following parameters: • Application limit – positive A-side current in [A] **Default value: 0.00 A** • Application limit – negative B-side current in [A] **Default value: 0.00 A**

26624737/EN – 03/2020

Operating modes

State of charge control

Parameter group	Description
State of charge control	The permitted operating range of the state of charge control is set using the following parameters. • Maximum voltage – operating range A • Maximum voltage – operating range B • Minimum voltage – operating range A • Minimum voltage – operating range B **Default value (each): 0 V** Setpoints (independent of the set unit) are limited to these values. **INFORMATION:** If control of the boost mode is enabled via the software module, the boost mode becomes active from the voltage "Minimum voltage – operating range A/B" (depending on the selected control side of the output stage).
Voltage lower limit axis operation	The voltage lower limit in axis operation is set using the following parameters: • Lower limit DC link voltage – axis operation **Default value: 0 V** The connected axes can be operated above this voltage. However, the value must be below the operating range.
Backup mode / remaining runtime calculation	The remaining runtime calculation is set using the following parameters: • Lower limit DC link voltage – backup mode **Default value: 0 V** Operation of the 24 V supply from the DC link is possible in case of a power failure up to this voltage. **Information:** The value must be below the "Voltage limit axis operation" value.
Setpoint/actual value representation	The representation of the setpoint/actual value is set using the following parameters: • Unit for state of charge: – **Electrical voltage [V]** – Operating range in percent [%] – Electrical energy [Ws] • Unit for charge limit: – **Electrical power [W]** – Electrical current [A]

26624737/EN – 03/2020

Energy storage discharge

Parameter group	Value
Discharging of energy storage units	The energy storage discharge is set using the following parameters: • External resistor – value Ohmic resistance of connected braking resistor according to data sheet. **Default value: 0 Ω** • External resistor – continuous power Continuous power that can be dissipated in heat in kW according to data sheet. **Default value: 0 kW** If the parameters are not set correctly (e.g. 0 Ω and 0 kW), the software module signals a fault and the operating mode cannot be used.

Energy storage unit

Parameter group	Value
Energy storage unit **Information:** Energy storage units are always entered in the electrostatic nominal capacitance. In addition, their installation location (A or B side of the corresponding device) is entered as well. Only MDC90A capacitor modules are parameterized via the "Fix" values.	Setting the energy storage unit and other capacities integrated in the application as the basis for calculations. Internal device capacities need not be taken into account. • DC connection energy storage unit can be disconnected - The energy storage unit can be disconnected from the DC link or the connected device on all poles via a corresponding contactor or load-break switch circuit. **Default value: No** • Electrostatic nominal capacitance - Capacitance of connected energy storage units (except fixed capacitor modules MDC90A). **Default value: 0 F** • Output stage side - Side of the output stage to which the nominal electrostatic capacitance is connected. **Default value: A-side** • Fixed capacitance application (A-side) - Capacitance of fixed MDC90A capacitor modules on the A-side of the output stage. **Default value: 0 mF** • Fixed capacitance application (B-side) - Capacitance of fixed MDC90A capacitor modules on the B-side of the output stage. **Default value: 0 mF**

Parameter group	Value
Multi-stage fuse configuration	Setting of the "multi-stage fuse" additional function via the following parameters: • Multi-stage fuse concept – activate **Default value: No** • Multi-stage fuse concept Switching threshold bridging on **Default value: 0 V** • Multi-stage fuse concept Switching threshold bridging off **Default value: 0 V** • Multi-stage fuse concept Sub-fuse nominal value **Information:** A permanent load capacity of the fuse of 80% of the nominal value is assumed and applied accordingly as a limit. **Default value: 0 A**

Fieldbus interface

Parameter group	Value
Fieldbus configuration	• Activate fieldbus connection – Fieldbus block is integrated and connected to the fieldbus handler of the MOVI-C® CONTROLLER. • Start address Start address of the fieldbus process data words in the array of the bus system. Counting starts at 1.
Process data length	• Process data length – Number of process data words used by the software module. **Default value: 8** • Unit for state of charge: – **Electrical voltage [V]** – Operating range in percent [%] – Electrical energy [Ws] • Unit for charge limit: – **Electrical power [W]** – Electrical current [A]

Module identification

Parameter group	Description
Module identification	Includes name and version for identifying the software module.

26624737/EN – 03/2020

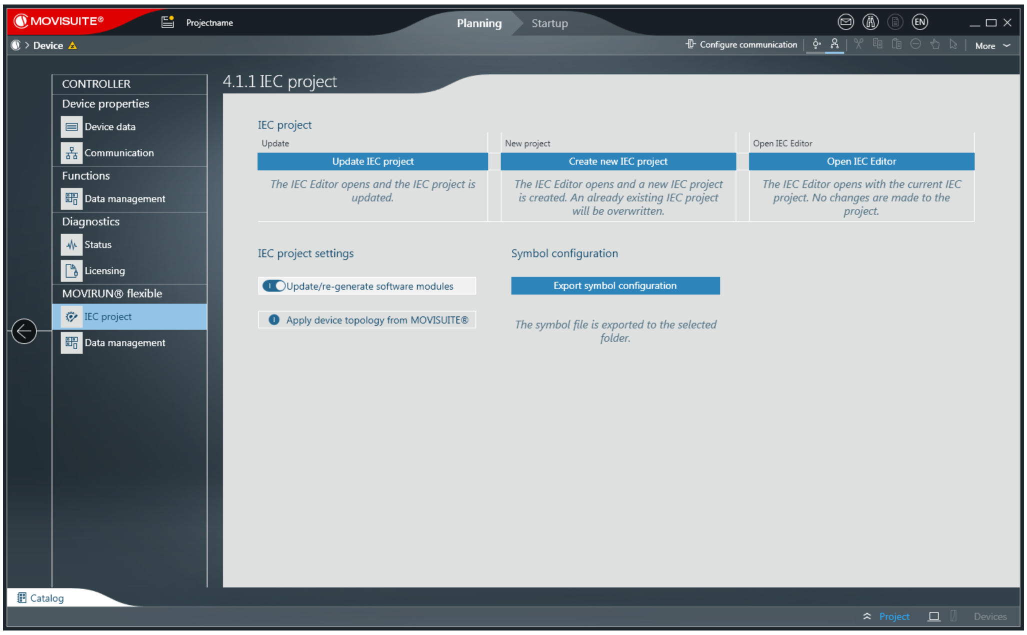
5.4 Generating an IEC project

Carry out the following steps to create an IEC project using automatic code generation and based on the configuration settings in MOVISUITE®.

✓ Configuration of the MOVISUITE® project has been completed.

1. In the function view of MOVISUITE®, click on the software module section of the MOVI-C® CONTROLLER.

   ⇨ The "IEC project" menu opens.



27021618448637067

INFORMATION

If you have carried out the configuration in MOVISUITE® using the "Startup" mode and the message "Device cannot be reached" appears, proceed as follows:

- If the MOVI-C® CONTROLLER is not available via the network, switch over to "Planning" mode.
- If the MOVI-C® CONTROLLER is available via the network, carry out a network scan and connect the MOVI-C® CONTROLLER in the network view with the MOVI-C® CONTROLLER in the function view.

2. Click [Create new IEC project].

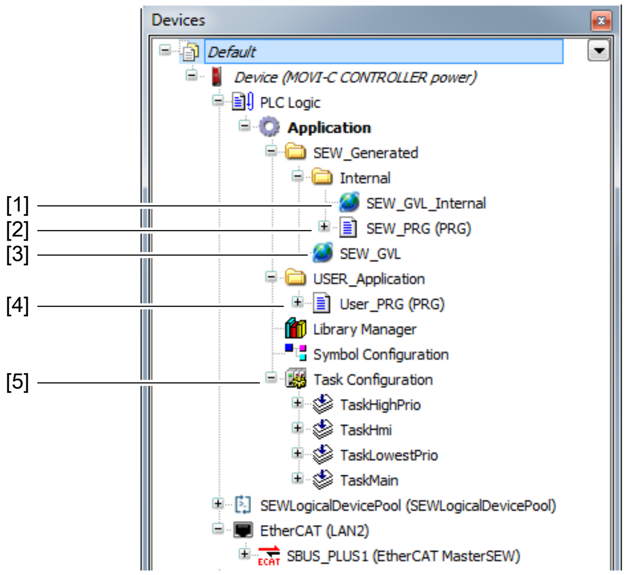
   ⇨ The IEC Editor opens and a new IEC project is created.

INFORMATION

If changes are made to the project structure, to inverter data sets, or to a software module configuration after the IEC project is generated for the first time, a notification symbol is displayed on the MOVI-C® CONTROLLER node. Click on the message icon for more information about the change, and to update the IEC project.

5.4.1 IEC project structure

The IEC project has the following basic structure:



18014423003085323

No.	Name	Description
[1]	SEW_GVL_Internal	The SEW_GVL_Internal global list of variables contains the instances that correspond to the software module used. These variables may not be written to from the user program. In addition, the structure contains an instance as a communication buffer for controlling or monitoring the software module by means of a monitor.
[2]	SEW_PRG	Program that contains all the important instance calls. Automatic code generation recreates this program in accordance with the configuration made in MOVISUITE® each time the IEC project is created, thereby overwriting the previous version. Therefore, you should not make any changes to this program.
[3]	SEW_GVL	The SEW_GVL global list of variables is the interface for accessing the software module features.
[4]	User_PRG	The user program is created once, initially, by automatic code generation. Since the program is not overwritten with each subsequent creation, this is the appropriate place for integrating user programs. The program is divided into five actions. These actions differ in the time at which they are called during the program sequence.
[5]	Task configuration	The list of tasks created in the project. Automatic code generation initially adds tasks that differ in how they are prioritized. The user can add additional programs to existing tasks or create new tasks. It is the responsibility of the user to design the capacity utilization of the tasks to enable the tasks to be processed within the required cycle time. Moving beyond the cyclical tasks, in particular, prevents setpoints for the interpolating axes from being generated in time, which means that these axes cannot be operated properly.

6 IEC programming

6.1 Opening the IEC project

- If an IEC project has already been generated, select the [IEC Editor] entry under "Tools" from the context menu of the MOVI-C® CONTROLLER in MOVISUITE®.
- If no IEC project has been generated, follow the steps described in the "Generating an IEC project" (→ 🗎 23) chapter.

6.2 User interface

The user interface is implemented in the IEC program by an instance in the global variable list *SEW_GVL*.

The following figure shows the interface in the IEC Editor:

Interface_MDP92A	SEW_MK_PES_PowerMode.EnergyHubPowerMode_UI	
xError	BOOL	FALSE
xWarning	BOOL	FALSE
udiMessageID	UDINT	16#00000000
sAdditionalText	STRING(Constants.gc_udiLengthAdditionalText)	''
xReset	BOOL	FALSE
xGetAccessControl	BOOL	TRUE
xControlActive	BOOL	FALSE
ModeControl	SEW_PES_IEnHubBas.ST_ModeControl	
Basic	SEW_PES_IEnHubCom.ST_Basic	
Inverter	SEW_PES_IEnHubCom.ST_Inverter	
RemainingRuntime	SEW_PES_IRemRuntimeCalc.ST_RemainingRuntime	
ModeStateOfChargeControl	SEW_PES_IModeSoCCtrl.ST_ModeSoCControl	
ModeStorageDischarge	SEW_PES_IModeStorDisch.ST_ModeStorageDischarger	
ModeDCSync	SEW_PES_IModeDCSync.ST_ModeDCSync	
StorageMonitoringASide	SEW_PES_IStorageMon.ST_StorageMonitor_A	
StorageMonitoringBSide	SEW_PES_IStorageMon.ST_StorageMonitor_B	
MultipleFuseCtrl	SEW_PES_IMultipleFuseCtrl.ST_MultipleFuseCtrl	

9007230385321227

26624737/EN – 03/2020

6.3 Basic functions

6.3.1 Diagnostics

Variables for reporting and writing errors and warnings.

Variable name	Description
xError	Data type – BOOL
	• TRUE – Error present • FALSE – No error present
xWarning	Data type – BOOL
	• TRUE – Warning present • FALSE – No warning present
udiMessageID	Data type: UDINT
	Message ID number
sAdditionalText	Data type: STRING
	Additional message text
xReset	Data type – BOOL
	• TRUE – Reset messages • FALSE – Do not reset messages

6.3.2 Access management

Variables for managing access permissions.

Variable name	Description
xGetAccessControl	Data type – BOOL
	• TRUE – Request access • FALSE – Return access
xControlActive	Data type – BOOL
	• TRUE – Access granted • FALSE – Access denied

Access via user interface (UserInterface):

An instance requests access by setting *xGetAccessControl* to "TRUE". If *xControlActive* returns a "TRUE" value, access has been granted and is now permitted.

Access via other function blocks in the user program:

If another function block in control mode wants to access the device interface at the same time as the user interface (UserInterface), the device interface decides which one receives write permission based on the priority of the function block. The user interface (UserInterface) has the highest priority. This means if the user interface (UserInterface) has control access, then *xControlActive* returns "FALSE" to all other function blocks.

26624737/EN – 03/2020

6.3.3 Operating mode selection (ModeControl)

The *ModeControl* structure is used to select the "operating mode" (→ 🗎 31) of the software module. Several operating modes cannot be activated at the same time because each operating mode is specifically defined to perform a certain function.

IN

Variable name	Description
uiRequestedMode	Data type - UINT
	Operating mode selection • 0 – No operating mode • 500 – State of charge control (StateOfChargeControl) • 510 – Energy storage discharge (StorageDischarger) • 520 - DC synchronization (DCSync)

OUT

Variable name	Description
uiActualMode	Data type: UINT
	Currently activated operating mode

6.3.4 Basic measured values (Basic)

The *Basic* structure contains variables for outputting basic measured values of the device.

OUT

Grid

The *Grid* structure contains variables to output information on the electrical supply system. The measured outer conductor voltages as well as additional information on the grid status and the status of the individual line phases are displayed.

INFORMATION

The information is only output if the connected device provides this information. Using MDE90A, for example, only the grid status is sent from the module bus master device because the device is not physically connected to the external supply system.

Variable name	Description
rActualVoltageL1L2	Data type – REAL
	Current phase-to-phase voltage (RMS) between phases L1 and L2 [V]
rActualVoltageL1L3	Data type – REAL
	Current phase-to-phase voltage (RMS) between phases L1 and L3 [V]

Variable name	Description
rActualVoltageL2L3	Data type – REAL
	Current phase-to-phase voltage (RMS) between phases L2 and L3 [V]
rEstimatedGridCurrent	Data type – REAL
	Calculated grid phase current per phase (AC RMS [A])
xMainsConnected	Data type – BOOL
	• TRUE – The measured voltage is within the permissible range (supply system is connected). • FALSE – The measured voltage is outside the permissible range (supply system is not connected).
xPhase1Connected	Data type – BOOL
	• TRUE – L1 phase voltage has expected value • FALSE – Phase voltage L1 failed
xPhase2Connected	Data type – BOOL
	• TRUE – L2 phase voltage has expected value • FALSE – Phase voltage L2 failed
xPhase3Connected	Data type – BOOL
	• TRUE – L3 phase voltage has expected value • FALSE – Phase voltage L3 failed

DCLink

The *DCLink* structure contains variables to output measured variables of the electrical DC link, such as currents, voltages, and power on the two control sides. The currently selected control side is also output.

Variable name	Description
eControlSideActive	Data type – ENUMERATION
	Active control side of the connected device • A_SIDE – A-side is control side for FCB 55. • B_SIDE – B-side is control side for FCB 55. • NONE – No side selected as control side. • BOTH – Both sides selected as control side.
rActualCurrent_A	Data type – REAL
	Actual DC current [A] on the A-side of the device
rActualCurrent_B	Data type – REAL
	Actual DC current [A] on the B-side of the device
rActualVoltage_A	Data type – REAL
	Actual DC voltage [V] on the A-side of the device
rActualVoltage_B	Data type – REAL
	Actual DC voltage [V] on the B-side of the device

26624737/EN – 03/2020

Variable name	Description
rActualPower_A	Data type – REAL
	Actual DC power [W] on the A-side of the device
rActualPower_B	Data type – REAL
	Actual DC power [W] on the B-side of the device

6.3.5 Inverter status (inverter)

The *Inverter* structure contains variables to output the status information of the device and to control the digital inputs and outputs.

IN

Variable name	Description
xActivateStandBy	Data type – BOOL
	• TRUE – Activate standby mode • FALSE – Deactivate standby mode
wDigitalOutputs	Data type - WORD
	Control of the digital outputs of the device **Information:** Depending on the software module used, digital outputs are already reserved and cannot be controlled in this case.

OUT

Variable name	Description
xConnected	Data type – BOOL
	• TRUE – Communication connection active • FALSE – Communication connection inactive
eInverterMode	Data type – ENUMERATION
	Active FCB or active operating mode. • FCB_STANDARD: 0 Default FCB • FCB_BRIDGE_DISABLED: 51 Output stage inhibit activated • FCB_BRAKE_CHOPPER_BLOCKED: 52 Brake chopper blocked due to fault • FCB_VOLTAGE_CONTROL: 55 DC link voltage control activated
wDigitalInputs	Data type - WORD
	State of the digital inputs **Information:** Depending on the software module used, digital inputs are already reserved and cannot be controlled in this case.

Variable name	Description
usiErrorID	Data type: USINT
	Error ID
usiErrorSubID	Data type: USINT
	Suberror ID
xLimitActive	Data type – BOOL
	• TRUE – An internal limit is active and limits the setting range of the device • FALSE – The device operates within its setting range
xPowered	Data type – BOOL
	• TRUE – The output stage of the device is enabled • FALSE – The output stage of the device is inhibited
xReady	Data type – BOOL
	• TRUE – The device is ready for operation • FALSE – The device is not ready for operation
xSetpointActive	Data type – BOOL
	• TRUE – Setpoints are processed • FALSE – Setpoints are not processed
xSetpointReached	Data type – BOOL
	• TRUE – Specified setpoint is reached. • FALSE – Specified setpoint is not reached.
xStandByActive	Data type – BOOL
	• TRUE – Standby mode activated. • FALSE - Standby mode deactivated.

6.3.6 Accessing measured values

The following code example shows how to access measured values via the respective OUT structures to use them, e.g. for additional calculations.

```
rPGrid   :=  Interface_MDP92A.Basic.OUT.DCLink.rActualPower_A;
rUDC    :=  Interface_MDP92A.Basic.OUT.DCLink.rActualVoltage_A;
rSoC    :=  Interface_MDP92A.Basic.OUT.DCLink.rActualVoltage_A*
            Interface_MDP92A.Basic.OUT.DCLink.rActualVoltage_A*
            REAL#1.9290E-06;
rPStorage := Interface_MDP92A.Basic.OUT.DCLink.rActualPower_A;
rPLoad   :=  rUDC * rUDC / REAL#13.2;
xError   :=  Interface_MDP92A.xError;
```

26624737/EN – 03/2020

6.4 Operating modes

6.4.1 State of charge control (StateOfChargeControl)

The *StateOfChargeControl* operating mode is used to adjust the power or current flow to a configured value using the connected device (e.g. MDP92A, MDE90A).

INFORMATION

With the MDP92A, the operating mode can work only on the A-side. With MDE90A, the control side can also be switched to the B-side to allow for specific charging of a connected energy storage unit.

INFORMATION

The operating mode automatically limits the setpoint for the state of charge to the specified "operating range" (→ 🗎 33). Additionally, the setpoints for the charge limit are automatically limited in such a way that the "application limits" (→ 🗎 18) for currents set in the device are not exceeded on the respective control side.

Operating principle

The state of charge to be adjusted is passed on as follows depending on the "configuration" (→ 🗎 20):

- As absolute voltage in [V] – Adjust the state of charge according to the specified absolute voltage.
- As percentage value (between 0% and 100%) – Adjust the state of charge depending on the operating range specified in the "configuration" (→ 🗎 20).
- As absolute energy value in [Ws] – Adjust the state of charge according to a specified absolute amount of energy in the DC link.

Furthermore, a charge limit can be transferred as follows to limit the charging of the DC link or connected energy storage unit:

- As power limit – Limit the device current on the respective control side so that the specified power limit results with the current voltage.
- As current limit – Limit the maximum device current on the respective control side to the specified value.

26624737/EN – 03/2020

Boost mode

In certain cases, it may be necessary for the device to automatically allow higher power or higher currents. This is possible by activating the boost mode in the device (for details on the boost mode, see the "Power and Energy Solutions" device manual). If the MOVI-C® CONTROLLER is to control the boost mode, it must be activated via *xActivateBoost* in the device and the minimum application limit must be set to "0 V". Depending on the operating status of the system, it may be necessary to initially deactivate the boost mode, e.g. when charging the energy storage unit for the first time, which should be done at low power.

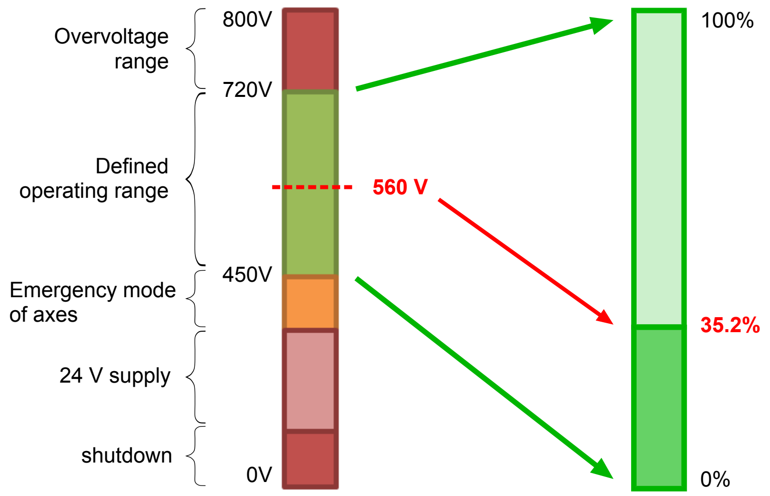
If the system is in normal operation, the boost mode can be activated via *xActivateBoost* and thus released again. If the voltage drops below the minimum voltage of the defined operating range due to high load (in the example in the following chapter, this is 450 V), the device releases the parameterized application limits of the power and currents regardless of the MOVI-C® CONTROLLER setting, so that a higher current/power flow may be possible to provide additional support for the DC link.

26624737/EN – 03/2020

Specifying the operating range

A valid operating range must be specified in the "Configuration" (→ 🗎 20) in order to use the operating mode. Setpoints to the device for the state of charge are always limited to this operating range, irrespective of the set unit. This can be a range between 450 V and 720 V, for example (see figure). This means that this voltage range corresponds to the state of charge from 0% to 100%.

The following figure shows the operating range to be defined:



The device operates primarily with the voltage of the DC link or the connected energy storage unit. For the capacitors used, this is an indicator of the amount of charge or energy in the DC link and consequently also of the state of charge.

The amount of energy in the energy storage unit or in the DC link is quadratically dependent on the current voltage according to the following formula:

$$E = \frac{1}{2} C U^2$$

In this way, the state of charge can be calculated according to the following formula using only the measured voltages and the defined operating range:

$$SoC\% = \frac{(U_i^2 - U_u^2)}{(U_o^2 - U_u^2)} \cdot 100\%$$

U_o and U_u define the upper and lower voltage limits of the operating range, and U_i the measured voltage at which the state of charge should be calculated. Similarly, this formula is used to convert state of charge setpoints into voltages and to make these voltages available to the device as voltage setpoints. The state of charge cannot be negative but can take on a value above 100%. This case occurs, for example, if the DC link adopts a higher voltage value than defined by the operating range when absorbing excess recuperation energy.

IN

The following input variables are available for using the operating mode.

Variable name	Description
xStart	Data type – BOOL
	• TRUE - Start function. • FALSE - Stop function.
xActivateBoostMode	Data type – BOOL
	• TRUE - Enable boost mode (requires an activated boost mode in the device). • FALSE - Block boost mode.
eControlSideRequest	Data type – ENUMERATION
	Selection of the control side of the connected device. • A_SIDE – Select A-side as control side for FCB 55. • B_SIDE – Select B-side as control side for FCB 55.
rSoCSetpoint	Data type – REAL
	Setpoint of the state of charge (observe configured unit)
rChargeLimit	Data type – REAL
	Setpoint of the charge limit (observe configured unit)

OUT

The status of the operating mode is output via the following output variables.

Variable name	Description
eControlSideActive	Data type – ENUMERATION
	Active control side of the connected device • A_SIDE – A-side is control side for FCB 55. • B_SIDE – B-side is control side for FCB 55. • NONE – No side selected as control side. • BOTH – Both sides selected as control side.
rActualOpRangePct	Data type – REAL
	Current stage of charge of the selected control side [%]
rActualVoltage	Data type – REAL
	Actual voltage of the selected control side [V]
rActualEnergy	Data type – REAL
	Actual amount of energy of the selected control side [Ws]

26624737/EN – 03/2020

<table>
<tr><th>Variable name</th><th>Description</th></tr>
<tr><td rowspan="2">eOperatingRange

Information: Correct "configura-tion" (→ 🗎 20) of the voltage limits is a pre-requisite for properly displaying the state of charge.</td><td>Data type – ENUMERATION</td></tr>
<tr><td>Present state of charge.
ABOVE
The actual state of charge is above the normal operating range. Risk of overload of the DC link or energy storage unit.
IN_RANGE
The actual state of charge is within the normal operating range.
BELOW_AXES_AVAIL
The actual state of charge is below the normal operating range. Connected axes can still be operated, even in re-duced operation if necessary.
BELOW_24V_SUPPLY
The state of charge is too low for operating axes. How-ever, the MDS 24 V power supply unit can supply the system with power.
BELOW_SHUTDOWN
The state of charge is so low that the system switches off within a short time.</td></tr>
</table>

26624737/EN – 03/2020

Activating DC link control

INFORMATION

The settings described below can be made everywhere in the program code. To reduce runtime in the *HighPrio* task, we recommend using the *Main* task to make the settings.

The following code example (software module used: MOVIKIT® DirectMode) shows how to set the DC link voltage to 650 V and a maximum charging power of 40 kW on the MDP92A power supply module.

Request access:

```
Interface_MDP92A.xGetAccessControl:=TRUE;
```

Activate voltage control:

```
Interface_MDP92A.ModeControl.IN.uiRequestedMode:=500;
Interface_MDP92A.ModeStateOfChargeControl.IN.xStart:=TRUE;
```

Setpoint for DC link voltage [V]:

```
Interface_MDP92A.ModeStateOfChargeControl.IN.rSoCSetpoint:=650;
```

Setpoint for maximum charging power [W]:

```
Interface_MDP92A.ModeStateOfChargeCon-
trol.IN.rChargeLimit:=40000;
```

Control page selection:

```
Interface_MDP92A.ModeStateOfChargeControl.IN.eControlSideRe-
quest:=SEW_MK_PES_DirectMode.SEW_PES_EnHubPowMode.SEW_PES_IEn-
HubCom.E_ControlSide.A_SIDE;
```

26624737/EN – 03/2020

6.4.2 Energy storage discharge (StorageDischarger)

The energy storage discharge operating mode (*StorageDischarger*) is used to discharge the DC link using a braking resistor connected to a MDP92A power supply module. The function block defines a maximum discharging capacity so that the specification of the connected braking resistor is not exceeded.

⚠ WARNING

Electric shock from a DC link capacity or energy storage unit that has not been fully discharged.

Severe or fatal injuries can occur.

- Check whether there is any residual voltage and, if necessary, completely discharge the DC link capacity or the energy storage unit as described in the relevant documentation.
- Observe the safety notes and the documentation of the energy storage unit in use.

INFORMATION

To activate the operating mode, you must specify a value for the braking resistor and its continuous power in the configuration menu "energy storage discharge" (→ 🗎 21).

Operating principle

The *StorageDischarger* operating mode discharges the DC link and stops automatically when the measured voltage drops below the target voltage. The function block sends a pulse width modulation signal (PWM signal) to the MOSFET (metal-oxide semiconductor field-effect transistor) of the braking resistor on the MDP92A power supply module so that the selected discharging capacity is not exceeded. The frequency of the signal is 1 Hz. From a certain point onward, the residual voltage is so low that the MOSFET is continuously activated and the discharging capacity decreases.

If the target voltage has been reached, the PWM signal is switched off and the operating mode signals back that the discharge process has been terminated using the "Target voltage reached" feedback. An estimated discharging time is displayed during discharging.

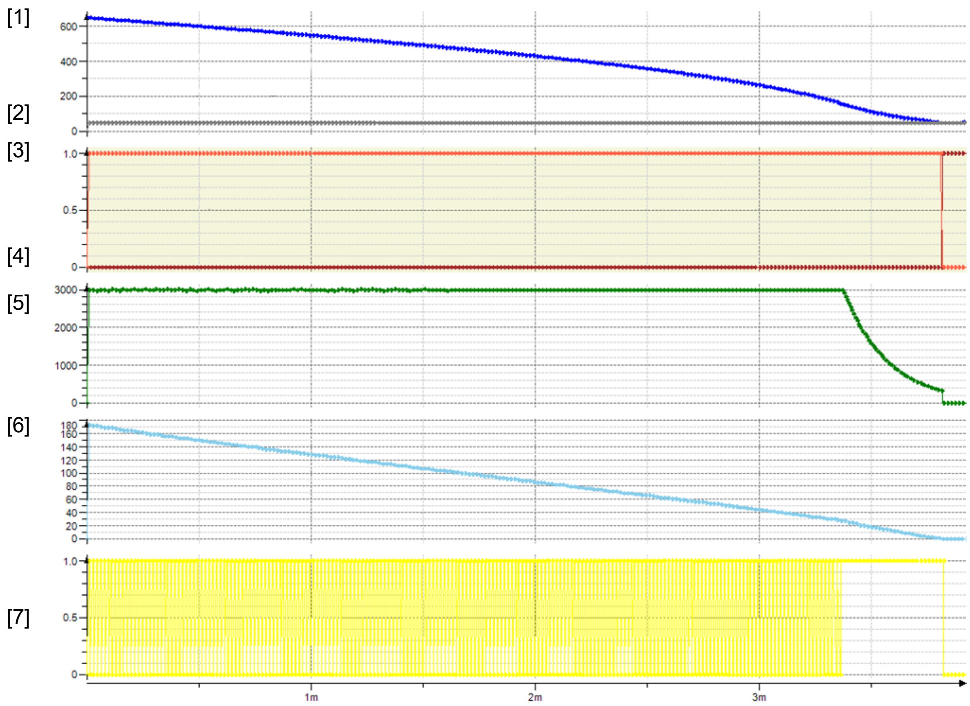
Special case: If 0 V is entered as target voltage and a residual voltage of approx. 1 V has been reached, the operating mode signals "Target voltage reached" as feedback. In this case, the braking resistor remains permanently connected to the DC link until the operating mode is no longer active (*xStart* = "FALSE").

INFORMATION

If "Target voltage reached" is signaled as feedback, the residual voltage has to be checked. Observe the safety note at the beginning of this chapter.

If the voltage is very high, or the resistance is very low, or the maximum continuous power of the connected resistor is very low, the smallest achievable PWM profile may cause the power in the resistor to increase more than intended for a given cycle time. A message is displayed if the currently achievable discharging capacity is at least 10% above the setpoint charging power.

The following figure shows a complete discharging process for the DC link. The DC link voltage is 650 V when the discharging process starts, the target voltage is 50 V and the maximum discharging capacity is 3 kW.



28798695307

[1] DC link voltage
[2] Discharging voltage
[3] Discharging active
[4] Discharging finished
[5] Current discharging capacity
[6] Remaining discharging time
[7] PWM profile

Explanation of the figure

The figure shows a constant discharging capacity of 3 kW, which decreases after slightly over 3 min, when the residual voltage is too low to produce 3 kW in the discharge resistor. From this point on, the PWM profile switches on the discharge resistor continuously. From the point in time at which the setpoint voltage is undershot, the PWM profile switches off again and signals "Target voltage reached" as feedback.

26624737/EN – 03/2020

IN

The following input variables are available for using the operating mode.

Variable name	Description
xStart	Data type – BOOL
	• TRUE - Start function. • FALSE - Stop function.
rVoltageSetpoint	Data type – REAL
	Target voltage for discharging the DC link in [V]

OUT

The status of the operating mode is output via the following output variables.

Variable name	Description
xDischargeActive	Data type – BOOL
	• TRUE – Discharging of the DC link is active. The braking resistor is controlled via the internal brake chopper. • FALSE – Discharging of the DC link is not active
xDischargeComplete	Data type – BOOL
	• TRUE – The target voltage for discharging the DC link has been reached. • FALSE – The target voltage for discharging the DC link has not been reached.
xHighPowerWarning	Data type – BOOL
	• TRUE – The braking resistor is overridden, meaning the current discharging capacity is at least 10% higher than the specified maximum capacity. • FALSE – The discharging capacity meets specifications.
rActualDischargePower	Data type – REAL
	Current discharging capacity [W]
rActualDCVoltage	Data type – REAL
	Current voltage of the DC link [V]
uiResidualDischarge-Time	Data type: UINT
	Estimated remaining time until the target voltage is reached [s]

26624737/EN – 03/2020

Discharging the DC link

The following application example describes the discharging of the DC link using the *StorageDischarger* operating mode.

⚠ WARNING

Electric shock from a DC link capacity or energy storage unit that has not been fully discharged.

Severe or fatal injuries can occur.

- Check whether there is any residual voltage and, if necessary, completely discharge the DC link capacity or the energy storage unit as described in the relevant documentation.
- Observe the safety notes and the documentation of the energy storage unit in use.

Preparation

If a fault message is present, the operating mode cannot be executed.

Possible causes for a fault message in the operating mode:

- The MDP92A power supply module signals a fault.
- The connected braking resistor signals a fault via DI 03 (not connected correctly or already too hot).
- The connected braking resistor has not been configured or has not been configured correctly in MOVISUITE® (e.g. resistance value ≤ 0 Ω or continuous power ≤ 0 kW)

In this case, identify the cause of the fault and rectify it. Afterwards, set the*xReset* variable to "TRUE" to reset the fault status. (Also possible in the case of device faults)

Discharge

✓ No fault message is present.

1. To select the *StorageDischarger* operating mode, set the *uiRequestedMode* variable to 510" (StorageDischarger) in the *IN* structure under *ModeControl*.
2. Enter a target voltage for the discharge of the DC link via *rVoltageSetpoint* in the *IN* structure under *ModeStorageDischarge*.
3. Set the *xStart* variable in the *IN* structure under *ModeStorageDischarge* to "TRUE".
   ⇨ If the current voltage in the DC link is at least 5 V above the setpoint voltage, the discharging of the DC link is started. To stop discharging, set the *xStart* variable to "FALSE".

⇨ If the target voltage was not reached, the *xDischargeComplete* variable displays the value "TRUE".

⇨ To restart discharging, the *xStart* variable must first be set to "FALSE" and then back to "TRUE".

6.4.3 DC synchronization

The *DC Synchronization* operating mode is used to disconnect and reconnect an energy storage unit from the DC link.

Operating principle

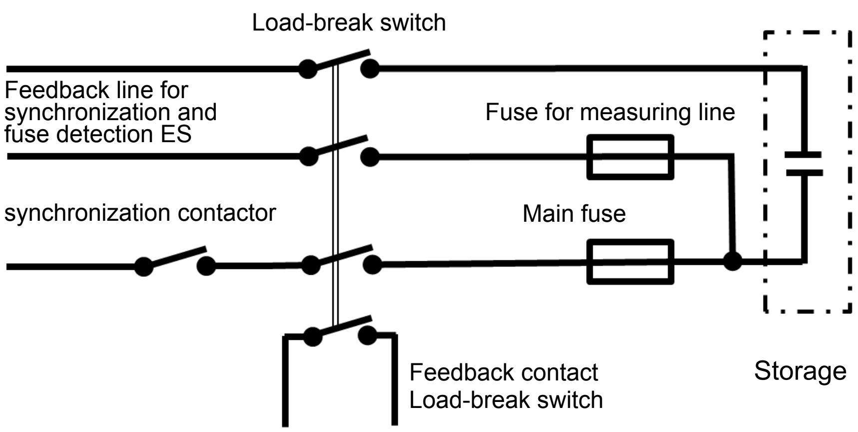
In order to disconnect or connect an energy storage unit from the DC link, a "synchronization contactor" is used, which must be installed in the supply line to the negative pole.

If the load-break switch is opened, e.g. to perform service work on the DC link, the MOVI-C® CONTROLLER automatically opens the synchronizing contactor. The storage can now be switched on again via the load-break switch after completion of the work. The final connection via the synchronizing contactor can be assumed by the "DC Synchronization" operating mode. In this process, the DC link voltage is first adjusted to the storage voltage in order to prevent any current flow during the switching process. The contactor is then closed so that the storage is now permanently connected to the DC link.

The operating mode can be activated only if the energy storage unit is explicitly selected to be disconnectable in the configuration of the software module. If the energy storage unit cannot be separated from the DC link but is permanently installed, the *xStorageConnected* output is always "TRUE".

INFORMATION

In order to be able to connect storage devices with any state of charge, the minimum voltage of the side on which the energy storage unit is located (parameter "Application limit – Minimum voltage A-side", "Application limit – Minimum voltage B-side" or both) must be set to 0 V in the "Monitoring functions" configuration menu in the "Limit values" (→ 🗎 18) submenu.



32314328587

26624737/EN – 03/2020

Open-loop control (IN)

Variable name	Description
xStart	Data type – BOOL
	• TRUE - Start function. • FALSE - Stop function.
xSyncAndConnect	Data type – BOOL
	Choice of whether the operating mode switches the energy storage device on or off. • TRUE - Synchronize and connect the connected energy storage unit when starting the operating mode if it was not previously connected to the DC link. • FALSE - Disconnect the connected energy storage unit from the DC link when starting the operating mode.

Status (OUT)

Variable name	Description
xStorageConnected	Data type – BOOL
	Status of switching on the energy storage unit • TRUE - The connected energy storage unit is connected to the DC link. • FALSE - The connected energy storage device is not connected to the DC link.
xSyncActive	Data type – BOOL
	Status of the synchronization of the connected energy storage unit. • TRUE - DC link synchronization active. • FALSE - DC link synchronization not active.

26624737/EN – 03/2020

Configuring control via digital inputs and outputs

As a prerequisite for connecting an energy storage unit, the synchronizing contactor must be controlled via a digital output. In addition, the feedback contact of the load-break switch must be queried via a digital input.

The following options are available to the user for this purpose (default setting for the digital output is DO 02 on the device for controlling the synchronizing contactor and the digital input DI 02 as input for the feedback contact of the load-break switch):

- Control the contactors via digital inputs/outputs of the MDE90A/MDP92A (default setting)
- Control the contactors via digital inputs/outputs of external terminals

If the user decides to use external terminals, the *IOConnectorDCSync* function block must be used as a block for connecting. The function block offers the following inputs and outputs, which must be connected directly to digital input and output terminals:

- Synchronizing contactor control contact:

  ```
  xDOContactorMinus : BOOL;
  ```

- Load-break switch feedback contact:

  ```
  xDIMainCircuitBreakerClosed : BOOL;
  ```

Control

(Software module used: MOVIKIT® PowerMode)

1. Create a local instance of the *IOConnectorDCSync* block to call and set the input and output variables.

   ⇨
   ```
   fbIOExchange : SEW_MK_PES_PowerMode.SEW_PES_EnHubPowMode.
   SEW_PES_ModeDCSync.IOConnectorDCSync;
   ```

2. Connect the local instance by calling up the *LinkIOExchange* method on MOVIKIT® PowerMode.

   ⇨
   ```
   EnergyHubPowerMode.LinkIOExchange(
   itfQueryInterfaceSEW:=fbIOExchange);
   ```

3. Add the two aforementioned input and output variables of the *fbIOExchange* block (*xDOContactorMinus*, *xDIMainCircuitBreakerClosed*) to the terminal used in the EtherCAT® mapping.

⇨ Control of the contactors is carried out automatically by the operating mode, so there is no need to call up the block again.

⇨ As soon as the *IOConnectorDCSync* is linked to the MOVIKIT®, the digital inputs and outputs of the unit are enabled again and can be used otherwise.

26624737/EN – 03/2020

Disconnecting the energy storage unit from the DC link

INFORMATION

To disconnect an energy storage unit from the DC link or the device, all connected axes at idle state and their output stages must be disabled.

If the idle state of all axes is ensured, the operating mode switches the connected device to FCB 51 (output stage inhibit) and opens the synchronization contactor. The energy storage unit can now be manually disconnected on all poles using the load-break switch. The *xStorageConnected* output is set to "FALSE". The process is now complete.

- Axes that are not connected via MOVIKIT® software modules must be brought to idle state before starting the operating mode. Monitoring by MOVIKIT® Power-Mode and the *DC Synchronization* operating mode do not occur in this case, and it is the responsibility of the programmer to ensure this.
- Axes that are operated via a MOVIKIT® software module of the "Motion" category can be made known to MOVIKIT® PowerMode directly via the EnergyHubPower-Mode.LinkAxis `EnergyHubPowerMode.LinkAxis(itfQueryInterfaceSEW := Axis1);` method invocation. `EnergyHubPowerMode` is the MOVIKIT® PowerMode function block and `Axis1` is the MOVIKIT® function block of the axis that implements the *IBasic* and *IInverter* interfaces. The *xStandstill* and *xPowered* bits of the axis are then called up automatically and the operating mode does not disconnect the energy storage unit until all axes report an idle state and output stage inhibit.

Control

✓ All connected axes are in an idle state and their output stages are disabled.

1. Set the *xStart* variable in the *IN* structure under *DC synchronization* to "TRUE" and the *xSyncAndConnect* variable to "FALSE".

   ⇨ When the energy storage unit is connected, it is disconnected from the DC link.

⇨ When all axes are in idle state, the operating mode switches the connected device to FCB 51 (output stage inhibit) and opens the synchronization contactor. The storage can now be disconnected by the load-break switch without load on all poles.

⇨ If the energy storage unit is disconnected from the DC link, the *xStorageConnected* variable returns "FALSE".

Synchronizing and switching on the energy storage unit

INFORMATION

When switching on, there is no explicit check on whether axes are in traverse mode at that time.

- To increase the service life of the contactors, make sure that no axes are in operation during synchronization.

1. Set the variables *xStart* and *xConnectStorage* in the *IN* structure under *DC synchronization* to "TRUE".
   ⇨ The algorithm for the synchronization of the energy storage unit starts.

Switch-on procedure

INFORMATION

The feedback contact of the load-break switch is continuously monitored as soon as the MOVI-C® CONTROLLER is ready for operation with the software module. If the user opens the load-break switch although the "DC Synchronization" operating mode has not been used for disconnection first, the operating mode changes to the "Storage not connected" state. The synchronization contactor is then opened. The mode reports "FALSE" via *xStorageConnected*.

Switching on the energy storage unit is carried out in the following steps:

Prerequisite: The feedback contact of the load-break switch signals that the load-break switch is closed (the corresponding digital input is at 24 V or "TRUE").

1. DC link voltage is adjusted to the voltage in the energy storage unit.
2. The synchronizing contactor is closed.

The storage is now connected to the DC link. The variable *xStorageConnected* is set to "TRUE". During synchronization, the activity is reported via the *xSyncActive* variable.

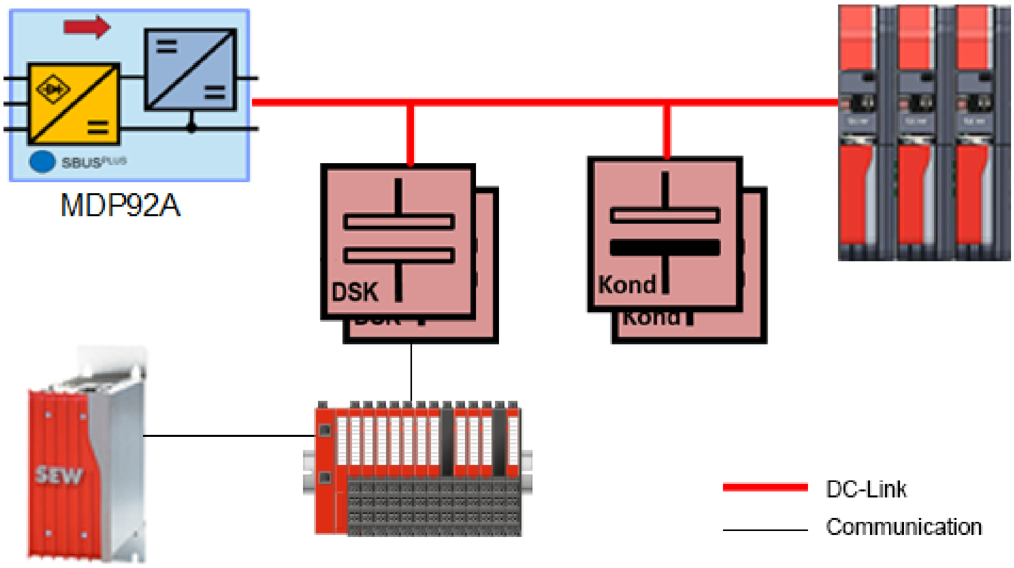
26624737/EN – 03/2020

6.5 Additional functions

6.5.1 StorageMonitor

The *StorageMonitor* function enables the evaluation of diagnostic interfaces of different energy storage types to protect the energy storage against overloading.

Energy storage units can be used that are either installed in the DC link without control or are connected to the DC link via MDE90A. The following figure shows an example topology:



18014427144387083

Operating principle

Some energy storage units have their own diagnostic interfaces. The diagnostic interfaces are different, but in most cases, they contain the following information:

- Overvoltage display (digital output)
- Overtemperature display (digital output)
- Overtemperature warning (not all)
- Temperature value (not all)
- Undertemperature display (not all)

Combined energy storage units are referred to as a "storage system" and are defined as follows:

- A storage system is a collection of energy storage units that form a common DC node.
- Each energy storage unit can provide its own status information. This status information is displayed in aggregated form for the storage system. Alternatively, system-wide status information can also be determined (using external measuring equipment).
- Only one storage system with any number of energy storage units can be installed in the DC link (power mode).
- In energy mode, multiple storage systems can be connected. However, a maximum of one storage system is connected to each MDE90A.

Integration

The storage diagnostic interfaces are mapped to blocks of the type *Storage* or *Storage_SEW_StandardSet*. The type *Storage_SEW_StandardSet* is derived from the type *Storage* and extends it with methods that simplify the configuration and the call. The code for the declaration of 2 storages of a storage system (software module used: MOVIKIT® PowerMode) looks, for example, as follows:

```
VAR

Storage_DcLink1 :
SEW_MK_PES_PowerMode.SEW_StorageMon.Storage;
Storage_DcLink2 :
SEW_MK_PES_PowerMode.SEW_StorageMon.Storage_SEW_StandardSet;

END_VAR
```

Alternatively, SEW-EURODRIVE's own storage (MOVI-DPS® or EKS storage unit) can be connected directly via the energy node. In this case, the energy storage unit is then configured via the energy node as well. Combining the two variants is also possible.

Configuration

The storage modules are located under *SEW_MK_PES_PowerMode.SEW_StorageMon*. Each storage module must first be initialized. The block of the type *Storage* has an init method, which must be called up once at the beginning. Look at the following example:

```
Storage_DcLink1.Init(
     rMaxTemperature := 60.0,
     rTemperatureWarnLevel := 55.0,
     rMinTemperature := -20.0,
     xIsOverTempAvailable := FALSE,
     xIsOverTempWarningAvailable := FALSE,
     xIsOverVoltageAvailable := TRUE,
     xIsTempValAvailable := TRUE,
     xIsUnderTempAvailable := TRUE,
     xIsWireBreakAvailable := TRUE,
     xStorageErrorAvailable := FALSE,
     sStorageName := 'Storage1'
     );
```

Explanation of the code example:

The xIs variables and *xStorageErrorAvailable* specify which signals the storage provides. If, for example, it does not supply an overtemperature signal but a temperature value, an overtemperature can still be detected. If the storage supplies a corresponding bit and value, the results are ORed.

The following variables are available for initializing the *Storage* function block:

Variable name	Description
rMaxTemperature	Data type – REAL
	Temperature limit for overtemperature
rMinTemperature	Data type – REAL
	Temperature limit for undertemperature

26624737/EN – 03/2020

Variable name	Description
rTemperatureWarnLevel	Data type – REAL
	Temperature limit for overtemperature warning
xIsTempValAvailable	Data type – BOOL
	• TRUE – Temperature value is available • FALSE – Value is not available
xIsOverTempAvailable	Data type – BOOL
	• TRUE – Overtemperature value is available • FALSE – Value is not available
xIsOverTemp-WarningAvailable	Data type – BOOL
	• TRUE – Overtemperature warning is available • FALSE – Value is not available
xIsUnderTempAvailable	Data type – BOOL
	• TRUE – Undertemperature limit is available • FALSE – Value is not available
xIsOverVoltageAvailable	Data type – BOOL
	• TRUE – Overvoltage limit is available • FALSE – Value is not available
xStorageErrorAvailable	Data type – BOOL
	• TRUE - Error signal is available. • FALSE - Error signal is not available
xIsWireBreakAvailable	Data type – BOOL
	• TRUE – Sensor wire break signal available • FALSE – Sensor wire break signal not available (sensor wire break is detected due to inconsistent measurement signal)

For certain energy storage units, there is a catalog initialization process that enters all Init parameters. This catalog initialization works only with certain combinations of IO bus modules and specific storage diagnostic interfaces. When using an LSMtron storage, the code looks like this:

```
Storage_DcLink1.Init_Catalog(
     eStorageType := SEW_MK_PES_PowerMode.
     SEW_StorageMon.SEW_PES_IStorageMon.
     E_StorageType.eLS_Mtron_081R0C_0100F_EA_YJ03,
     sStorageName := 'Storage2'
     );
```

The energy storage units must also be connected to the respective software module via the following call. The MDP92A is the function block of the type *EnergyHubPowerMode* integrated by automatic code generation.

```
MDP92A.LinkStorage(xIsDcLink := TRUE, itfQueryInterfaceSEW :=
Storage_DcLink1);
```

```
MDP92A.LinkStorage(xIsDcLink := TRUE, itfQueryInterfaceSEW :=
Storage_DcLink2);
```

Variable name	Description
itfQueryInterfaceSEW	Data type - IQueryInterfaceSEW
	Interface for linking a *Storage* or *Storage_SEW_Standard-Set* function block
xIsDcLink	Data type – BOOL
	• TRUE – The energy storage unit is connected in the DC link (A-side) • FALSE – The energy storage unit is connected separately (B-side of MDE90A)

SEW-EURODRIVE EKS Energy storage unit

The diagnostic interface of SEW-EURODRIVE's own EKS energy storage units can be read directly from the respective device (MDP92A, MDE90A) so that no additional I/O modules are required.

Proceed to monitor and transmit the temperature as follows:

✓ The diagnostic interface is connected to the respective device according to the electrical wiring diagram (see "Power and Energy Solutions" product manual).

1. In MOVISUITE®, open the "Monitoring functions" > "External error signals" configuration menu in the energy storage unit configuration.
2. Activate the monitoring of the external error signals.
3. Open the "Monitoring functions" > "Temperature" configuration menu.
4. Select the sensor type (e.g. EKS A) in the "Analog temperature sensor input" area.
   ⇨ The temperature sensor can be connected directly to the MDP92A/MDE90A.
5. Set the minimum temperature, the maximum temperature and a hysteresis in the "Temperature monitoring" section.
6. Activate the function in the "Temperature monitoring" section.

⇨ The diagnostic interface is automatically linked to the additional function, *Storage-Monitor*. Additional programming is not required and the diagnostics data is available to the user on the MOVI-C® CONTROLLER.

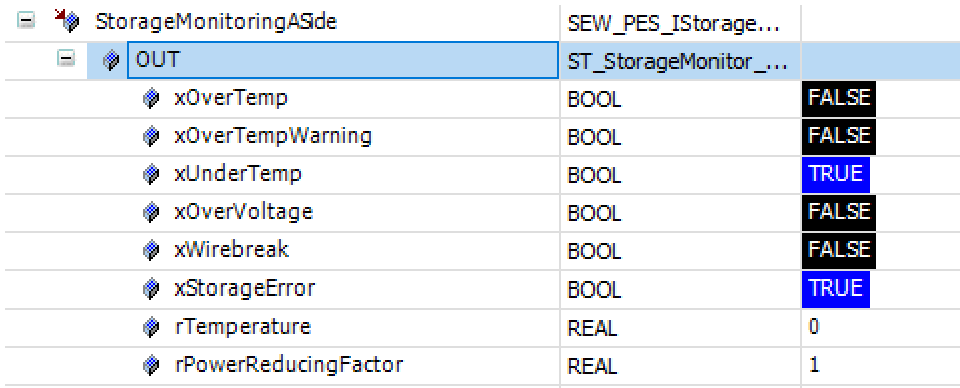
⇨ For the MDP92A device, the energy storage unit is displayed in the StorageMonitoringASide, as it can be located only on the A-side of the device.

⇨ In the MDE90A, the energy storage unit is displayed in the StorageMonitoringB-Side, as the storage is located on the B-side of the device in energy mode.

7. Update the configuration data of the MOVI-C® CONTROLLER

⇨ The changed parameters can be read out from the device and imported into the software module.

⇨ The storage can be read out via the following user interface:



32307552907

Explanation of the example:

The storage via *xStorageError* a general storage error (in case of the SEW-EURODRIVE EKS A, an overvoltage), as well as an undertemperature, since the measured value 0 °C was assumed to be too low for operation here.

The SEW-EURODRIVE EKS storage unit transmits a temperature value that represents the maximum of all temperatures measured in the connected energy modules. There is also wire break monitoring and an error bit that indicates overvoltage.

The control system evaluates the temperature and additionally issues error and warning messages as soon as the temperature has risen above a maximum threshold or fallen below a minimum threshold. A hysteresis value is available for monitoring the maximum temperature, so that a prewarning is triggered if the maximum temperature minus the hysteresis value is exceeded.

Operation

To supply the *Storage* block with values, its *AddStorageStatus* function must be called up cyclically. Cyclic usually means time discrete. So either the *CallHighPrio* task or the *ReadActValues* is suitable. If monitoring is less time-critical, the function can also be called in the *CallMain* task. A fixed time reference is not absolutely necessary.

The call-up in the cyclic function could look like the following example. To avoid misunderstandings, it is recommended to set ineffective variables to "FALSE".

```
Storage_DcLink1.AddStorageStatus(
     rInTemperature := rTemperature1,
     xInOvertemperature := FALSE,
     xInUndertemperature := FALSE,
     xInOverVoltage := xODI_OverVolt1,
     xInWireBreak := xODI_WireBreak1,
     xInStorageError := FALSE
     );
```

The *AddStorageStatus* function provides the following variables to transfer current diagnostic data of a storage interface:

Variable name	Description
xInOvertemperature	Data type – BOOL
	• TRUE – Temperature limit for overtemperature exceeded • FALSE – Temperature value within limits

26624737/EN – 03/2020

Variable name	Description
xInUndertemperature	Data type – BOOL
	• TRUE – Temperature limit for overtemperature not reached • FALSE – Temperature value within limits
xInOverVoltage	Data type – BOOL
	• TRUE – Overvoltage limit exceeded. • FALSE – Voltage value within limits
xInWireBreak	Data type – BOOL
	• TRUE - Wire break detected. • FALSE - No wire break detected.
xInStorageError	Data type – BOOL
	• TRUE - Error in energy storage unit. • FALSE - No error in energy storage unit.
rInTemperature	Data type – REAL
	Current temperature

If the *Storage* block is of the type *Storage_SEW_StandardSet*, then additional *AddStorageStatus* function variants are available. Instead of *AddStorageStatus*, for example, the function *AddStorageStatOCE11_On_LSMtron* can be called up:

```
Storage_DcLink2.AddStorageStatOCE11_On_LSMtron(
     xInOverVoltage := xODI_OverVolt,
     uiTemperature := uiODA_Temperature
     );
```

In this case, a SEWEURODRIVE IO system of type OCE11C is used to connect an LSMtron storage unit. The input values do not have to be processed; the temperature from the *AddStorageStatus* is expected as a floating-point value. The function takes the original analog value of the type INT or UINT and converts it so that the input values do not need to be further processed. However, this procedure is available only for an intended fixed combination of IO system and storage type.

There are *AddStorageStat* functions only for OCE11C IO systems with LSMTron and MOVI-DPS® (EKS storage unit) storages. In addition, there is the variant *AddStorageStatOCE11_On_PTnSensor*. The PTn sensor variant can work with the integer temperature values from the OAI45C IO module and converts the value internally. This variant is not linked to a specific type of storage unit, but to an analog input module that must be configured as a temperature input.

The following table shows the key data of the supported analog module for temperature:

Type	Module type	Number of inputs	Range of values	Additional info
OAI41C	Voltage measurement	4	0 − 10 V or ±10 V	Voltage value can be determined using conversion factors

26624737/EN – 03/2020

Type	Module type	Number of inputs	Range of values	Additional info
OAI45C	Resistance and temperature	4	0-3000Ω or PT100, PT1000, NI100, NI1000	Temperatures values are delivered directly

The OAI41C voltage measuring module is recommended for MOVI-DPS® or the SEW-EURODRIVE EKS energy storage unit.

Usage

INFORMATION

The storage monitoring is integrated into the regular error handling. Accordingly, storage problems also produce error messages in the software module.

The diagnostic data can be called up from the user interface of the energy node during operation. There is then one output for the A-side (DC link) and one for the B-side.

Interface in the IEC Editor

StorageMonitoringASide	SEW_PES_IStorageMon.ST_StorageMonitor_A	
OUT	ST_StorageMonitor_OUT	
xOverTemp	BOOL	FALSE
xOverTempWarning	BOOL	FALSE
xUnderTemp	BOOL	FALSE
xOverVoltage	BOOL	FALSE
xWirebreak	BOOL	FALSE
xStorageError	BOOL	FALSE
rTemperature	REAL	47.3
rPowerReducingFactor	REAL	0
StorageMonitoringBSide	SEW_PES_IStorageMon.ST_StorageMonitor_B	
OUT	ST_StorageMonitor_OUT	
xOverTemp	BOOL	FALSE
xOverTempWarning	BOOL	FALSE
xUnderTemp	BOOL	FALSE
xOverVoltage	BOOL	FALSE
xWirebreak	BOOL	FALSE
xStorageError	BOOL	FALSE
rTemperature	REAL	44.8
rPowerReducingFactor	REAL	0

32319475851

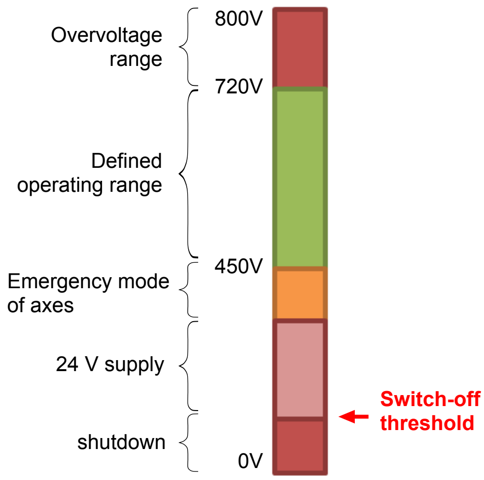
26624737/EN – 03/2020

6.5.2 RemainingRuntimeCalculator

The *RemainingRuntimeCalculator* function is a help function for calculating the remaining system runtime after a power failure. This function determines the estimated remaining time until the voltage in the DC link reaches the value as of which the MDS 24 V power supply unit can no longer provide 24 V supply from the energy storage unit.

Operating principle

If the voltage in the DC link reaches a value that falls below the "24 V supply" voltage range (see figure), the MDS 24 V power supply unit cannot convert the voltage into 24 V. As a result, the electronics of the control cabinet that is supplied by the MDS 24 V power supply unit is switched off completely. In case of a power failure, the remaining runtime (e.g. for data backup) until the switch-off threshold is reached is calculated by the *RemaininGRunTimeCalculator* on the assumption that there is a continuous power consumption and, therefore, an almost linear energy flow at the DC link. If no power failure was detected, the value for the remaining runtime is 0 ms.



Configuration

A *Lower limit DC link voltage – backup mode* must be configured in the parameter group "Backup mode / remaining runtime calculation" in the configuration menu "State of charge control" (→ 🗎 20) so that a valid remaining system runtime can be calculated by the *RemainingRuntimeCalculator* function. Enter a voltage value that is adapted to the existing hardware and up to which the 24 V supply is ensured.

Status variables (OUT)

Structure for displaying the remaining system runtime.

Variable name	Description
udiRemainingRuntimeMS	Data type: UDINT
	Remaining runtime of the 24 V supply until the switch-off threshold is reached in [ms].

26624737/EN – 03/2020

6.5.3 MultipleFuseControl

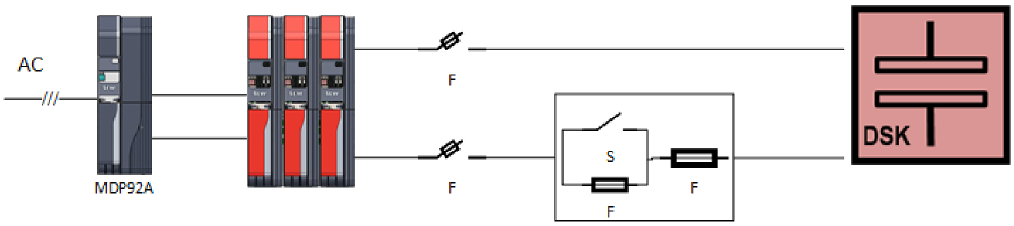
The *MultipleFuseControl* function controls the multi-stage fuse switch. Configuration and activation of the function takes place in MOVISUITE®.

Operating principle

Energy storage units based on double layer capacitor technology can have high internal resistances. At low voltage, a short-circuit current that occurs may consequently not be large enough to trigger the corresponding fuse in the supply line. Therefore, a smaller fuse is switched on or off if necessary. The software module takes over this control automatically. If *xCurrentLimitActive* is set to "TRUE", the current supplied by the device (MDP92A , MDE90A) is automatically limited to 80% of the maximum permissible current of the smaller fuse.

INFORMATION

The limitation of the current by the function must be considered in the application. If current limiting is active, there should be no travel operation.



32317207563

INFORMATION

If *xMultipleFuseCtrlActive* is set to "TRUE", digital input DI 01 and digital output DO 01 in the MDP92A/MDE90A are preassigned for controlling the multi-stage fuse circuit. As such, they cannot be used by the structures *Inverter.IN* and *Inverter.OUT*.

Status variables (OUT)

Variable name	Description
xMultipleFuseC-trlActive	Data type – BOOL
	• TRUE - Multi-stage fuse monitoring is active. • FALSE - Multi-stage fuse monitoring is not active.
xCurrentLimitActive	Data type – BOOL
	• TRUE - Feed current limited to the set value. • FALSE - Feed current not limited by the function.

26624737/EN – 03/2020

6.6 IEC libraries

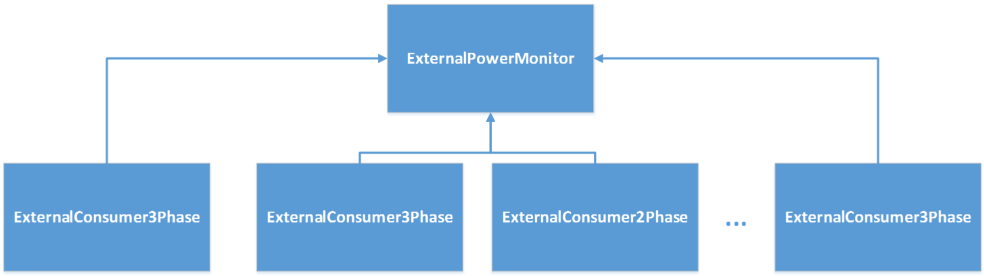
6.6.1 SEW PES GridMonitor

Energy loads in the control cabinet that are supplied via an additional AC supply instead of the DC link can be measured using power measurement terminals, for example. The IEC library *SEW PES GridMonitor* provides the function block *ExternalPowerMonitor* as well as sub-blocks for connecting AC loads.

Operating principle

The *ExternalPowerMonitor* function block serves as a central node. The *ExternalConsumer* function block is a standardized interface for connecting AC loads to the *ExternalPowerMonitor* function block. Up to 10 AC loads can be connected to the *ExternalPowerMonitor* function block.

The following figure illustrates this functional principle:

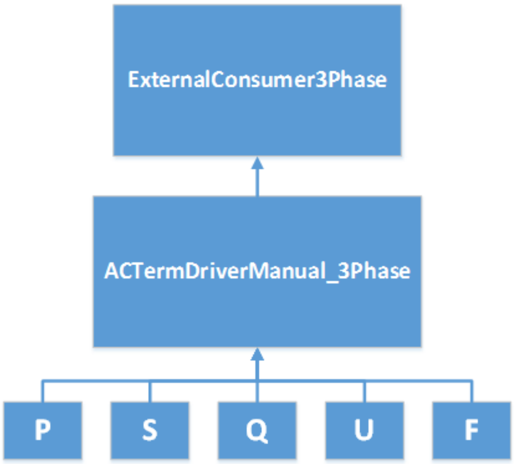


28625588875

The connected (2-phase or 3-phase) AC loads send data via a standardized interface to the *ExternalPowerMonitor* function block, which aggregates the data and displays it in a central location (e.g. determination of the total consumption).

The AC loads are connected via measurement terminals. Each consumption block (*ExternalConsumer2Phase* or *ExternalConsumer3Phase*) requires a connection block (driver). These connection blocks are available for selected power measurement terminals in the *SEW_PES_ACTerminalDrivers* IEC library to ensure quick and easy connection. Currently, the IEC library only offers the BECKHOFF EL3403 power measurement terminal.

The following example shows the *ACTermDriverManual_3Phase* connection block, which can be used if the measurement is performed using non-supported measurement terminals. This connection block provides arrays with data type REAL so that the programmer can transfer the values of the individual phases measured by a terminal to these variables. The values are subsequently forwarded and processed automatically.



31310434443

```
FUNCTION_BLOCK ACTermDriverManual_3Phase
EXTENDS ACTermDriverBasic3Phase
VAR_INPUT
     arApparentPower : ARRAY[0..2] OF REAL;
     arActivePower : ARRAY[0..2] OF REAL;
     arReactivePower : ARRAY[0..2] OF REAL;
     arVoltage : ARRAY[0..2] OF REAL;
     arFrequency : ARRAY[0..2] OF REAL;
END_VAR
```

ExternalPowerMonitor

The *ExternalPowerMonitor* function block represents the active power of all connected AC loads as a measured value and the sum of all active powers of the connected AC loads.

Variable name	Description
arActivePowerSingle	Data type: ARRAY OF REAL
	Active power measured values [W] of the connected AC loads. Up to 10 loads can be monitored using the "ExternalPower-Monitor" function block.
rActivePowerSum	Data type – REAL
	Total active power of all connected loads in [W]

ExternalConsumer3Phase

The *ExternalConsumer3Phase* function block provides the following electrical measured variables of a connected 3-phase AC load. It also indicates whether the connected loads measure and transmit these variables at all, i.e. whether these values are valid.

Variable name	Description
rApparentPower	Data type – REAL
	Apparent power measured value of the connected AC load in [VA]
xApparentPowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rActivePower	Data type – REAL
	Active power measured value of the connected AC load in [W]
xActivePowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rReactivePower	Data type – REAL
	Reactive power measured value of the connected AC load in [VAr]
xReactivePowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rFrequency	Data type – REAL
	Frequency measured value of the connected AC load in [Hz]
xFrequencyValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rVoltageL1N	Data type – REAL
	Voltage measured value L1-N of the connected AC load in [V]
xVoltageL1NValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rVoltageL2N	Data type – REAL
	Voltage measured value L2-N of the connected AC load in [V]
xVoltageL2NValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid

Variable name	Description
rVoltageL3N	Data type – REAL
	Voltage measured value L3-N of the connected AC load in [V]
xVoltageL3NValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid

ExternalConsumer2Phase

The *ExternalConsumer2Phase* function block provides the following electrical measured variables of a connected 2-phase AC load. It also indicates whether the connected loads measure and transmit these variables at all, i.e. whether these values are valid.

Variable name	Description
rApparentPower	Data type – REAL
	Apparent power measured value of the connected AC load in [VA]
xApparentPowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rActivePower	Data type – REAL
	Active power measured value of the connected AC load in [W]
xActivePowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rReactivePower	Data type – REAL
	Reactive power measured value of the connected AC load in [VAr]
xReactivePowerValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rFrequency	Data type – REAL
	Frequency measured value of the connected AC load in [Hz]
xFrequencyValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid
rVoltage	Data type – REAL
	Voltage measured value of the connected AC load in [V]

Variable name	Description
xVoltageValid	Data type – BOOL
	• TRUE – Measured value valid • FALSE – Measured value invalid

SetConfig methods

Methods for transferring settings such as a filter time constant for measured value filtering to the function block.

ExtConsumer3Phase

Method for transferring modified filter time constants for measured value filtering to the function block. The method must be called once. The filters are reinitialized automatically.

If the new filter time constants have been applied and the filters have been reinitialized, the method returns the BOOLEAN value "TRUE." Otherwise, the value is "FALSE." This allows you to check for correct reinitialization.

Variable name	Description
uiPriorityTask_CycleTime	Data type: UINT
	Sampling cycle of the priority task in [ms]
rTimeConstActivePowerPT1	Data type – REAL
	PT1 filter time constant of the active power in [s]
rTimeConstReactivePowerPT1	Data type – REAL
	PT1 filter time constant of the reactive power in [s]
rTimeConstApparentPowerPT1	Data type – REAL
	PT1 filter time constant of the apparent power in [s]
rTimeConstFrequencyPT1	Data type – REAL
	PT1 filter time constant of the frequency in [s]
rTimeConstVoltageL1NPT1	Data type – REAL
	PT1 filter time constant of voltage L1-N in [s]
rTimeConstVoltageL2NPT1	Data type – REAL
	PT1 filter time constant of voltage L2-N in [s]
rTimeConstVoltageL3NPT1	Data type – REAL
	PT1 filter time constant of voltage L3-N in [s]

ExtConsumer2Phase

Method for transferring modified filter time constants for measured value filtering to the function block. The method must be called once. The filters are reinitialized automatically.

If the new filter time constants have been applied and the filters have been reinitialized, the method returns the BOOLEAN value "TRUE." Otherwise, the value is "FALSE." This allows you to check for correct reinitialization.

Variable name	Description
uiPriorityTask_CycleTime	Data type: UINT
	Sampling cycle of the priority task in [ms]
rTimeConstActivePowerPT1	Data type – REAL
	PT1 filter time constant of the active power in [s]
rTimeConstReactivePowerPT1	Data type – REAL
	PT1 filter time constant of the reactive power in [s]
rTimeConstApparentPowerPT1	Data type – REAL
	PT1 filter time constant of the apparent power in [s]
rTimeConstFrequencyPT1	Data type – REAL
	PT1 filter time constant of the frequency in [s]
rTimeConstVoltagePT1	Data type – REAL
	PT1 filter time constant of the voltage in [s]

Measuring AC loads

Without ExternalPowerMonitor

The following application example shows how an AC load can be measured using measurement terminals. In this case, third-party hardware is used and the terminals have already been connected and the individual measured variables have already been calculated.

1. Create an instance of the “manual” terminal driver:

   ⇨
   ```
   FBDriverManual:
   SEW_MK_PES_DirectMode.SEW_GridMon.
   SEW_ACTermDrivers.ACTermDriverManual_3Phase;
   ```

2. Create an instance of the *ExternalConsumers3Phase* function block:

   ⇨
   ```
   FB3PhaseConsumer:
   SEW_MK_PES_DirectMode.SEW_GridMon.ExtConsumer3Phase;
   ```

3. Connect the function blocks in the *Init* action in the *User_PRG* program using the *LinkACDriver()* method in *ExternalConsumer*:

   ⇨
   ```
   FB3PhaseConsumer.LinkACDriver(
   itfACDriver:=FBDriverManual
   );
   ```

4. Initialize both function blocks in the *Init* action in the *User_PRG* program via their respective initialization methods as follows:

   ⇨
   ```
   FBDriverManual.Init(
   xApparentPowerValid:=TRUE,
   xActivePowerValid:=TRUE,
   xReactivePowerValid:=FALSE,
   xVoltageValid:=TRUE,
   xFrequencyValid:=TRUE);
   ```

26624737/EN – 03/2020

⇨ The "manual" terminal driver is provided with information about which of the measured values must be measured. In the example, these are the apparent power, the active power, the voltage and the frequency. You can disable measurement of the reactive power by setting the *xReactivePowerValid* value to the value "FALSE."

5. Initialize the *ExternalConsumer* function block with the cycle time of the calling task:

   ⇨
   ```
   FB3PhaseConsumer.Init(
   uiPriorityTask_CycleTime:=gc_uiTaskCycleTime
   );
   ```

6. Call the following method in the *ReadActualValues* action in the *User_PRG* program:

   ⇨
   ```
   FB3PhaseConsumer.ReadActualValues();
   ```

7. Call the following method in the *HighPrio* action in the *User_PRG* program:

   ⇨
   ```
   FBDriverMannual.CallHighPrio();
   ```

8. Write the measured values to the variables of the driver block:

   ⇨
   ```
   FBDriverMannual.arActivePower[0]:=rPGrid*2;
   FBDriverMannual.arActivePower[1]:=rPGrid*1.5-3000;
   FBDriverMannual.arActivePower[2]:=rPGrid*0.5+3000;
   ```

   ⇨ The *rPGrid* variable was copied with different scaling to the 3 active power inputs. The measured values can then be viewed centrally in the *ExternalConsumer* block.

The connection of the terminal may differ slightly if a supported terminal is used with a driver from the library. However, the method calls and the connections always take place as described in this chapter.

With ExternalPowerMonitor

The following application example (software module used: MOVIKIT® DirectMode) shows how several AC loads can be detected and totaled using the *ExternalPowerMonitor* block. This requires an additional instance of the *ExternalPowerMonitor*.

1. Create an additional instance of the *ExternalPowerMonitor* function block:

   ⇨
   ```
   FBAcPowerMonitor:SEW_MK_PES_DirectMode.SEW_GridMon.Extern-
   alPowerMonitor
   ```

2. Use an interface link to link the function blocks in the *Init* action in the *User_PRG* program to each *ExternalConsumer* function block

   ⇨
   ```
   FBAcPowerMonitor.LinkExtConsumer(
   itfExtConsum:=FB3PhaseConsumer
   );
   ```

3. Call the following method in the *ReadActualValues* action in the *User_PRG* program:

   ⇨
   ```
   FBAcPowerMonitor.ReadActualValues();
   ```

4. Call the following method in the *HighPrio* action in the *User_PRG* program:

   ⇨
   ```
   FBAcPowerMonitor.CallHighPrio();
   ```

⇨ The measured values of the AC load are called and processed directly by the *ExternalPowerMonitor* function block. The totaled power values can be queried centrally here.

26624737/EN – 03/2020

6.6.2 SEW PES PowerMonitor

INFORMATION

The module must be implemented manually in the project and is not integrated via automatic code generation.

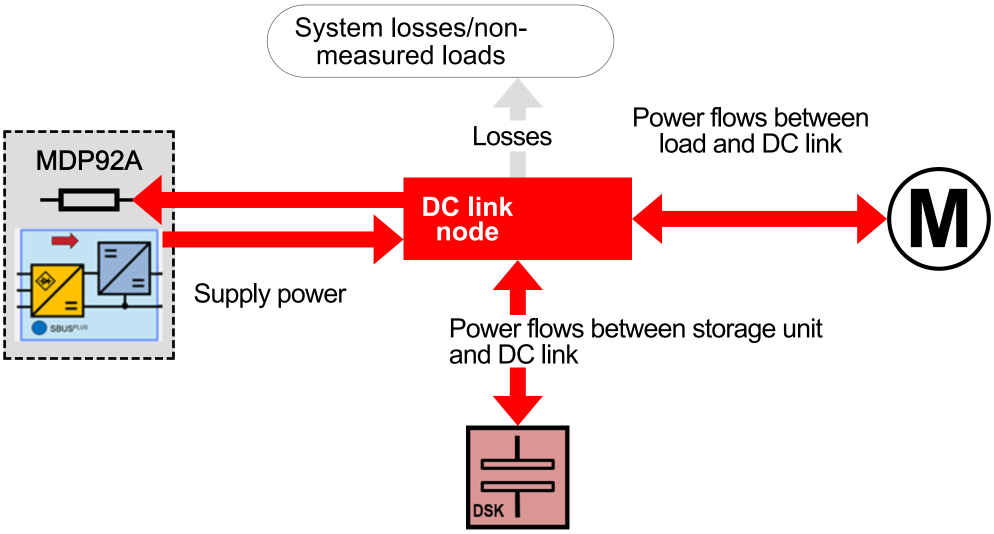
The *SEW PES PowerMonitor* IEC library systematically groups incoming and outgoing instantaneous power values.

Operating principle

When combining incoming and outgoing instantaneous power, a distinction is made between 3 variables:

- Grid power: Power delivered by the supply unit.
- Load power: Total power consumed in the system (in the same DC link).
- Storage unit power: Power that flows in or out of the additional storage unit.

All power values can also be negative. Axes can recuperate or a supply unit can transfer excess power from the DC link to the outside (conversion into heat at the discharge resistor).



9007228415314315

Integration

Do the following to integrate the functionality:

✓ For example, if MOVIKIT® MultiMotion is used, at least 2 process data words must be available for the actual values in the process data profile. For these process data words to appear in the actual values, the number of process data words must be increased manually by 2 in the configuration of the axis in the menu [Actual values] > [PI data]. The "Active electrical motor power" parameter (index: 8364.212) is then inserted into the process data at the actual values (low word and high word).

1. In MOVISUITE®, open the "Basic settings" configuration menu of the axis and select "InterpolationFlexFull" as the data module. This setting ensures that sufficient process data is available. Perform this step for each linked axis.

INFORMATION

Do not initialize the settings via the [Initialize settings] button after selecting the data module!

2. Generate an IEC project based on this configuration. See chapter "Generating an IEC project" (→ 🗎 23).
3. Create an instance of the *PowerMonitor* function block.
   ⇨ `fbPowerMonitor:SEW_MK_PES_DirectMode.SEW_PwrMon.PowerMon-itor;`
4. Link the MDP92A or MDE90A axes using *LinkEnergyDevice*.
   ⇨
   ```
   fbPowerMonitor.LinkEnergyDevice(
   xIsSupply:=TRUE,
   itfDevice:=MDP92A);
   ```
5. Link the SEW MultiMotion axes using *LinkMuMoAxis*.
   ⇨
   ```
   fbPowerMonitor.LinkMuMoAxis(
   itfAxisProcessData:=LogicalDevice_Axis1,
   uiPD_PowerIndex:=14, //Power has been mapped to PI15 and
   PI16
   uiPriorityTask_CycleTime:=gc_uiTaskCycleTime);
   ```
6. Call up the following function in the *CallHighPrio* task:
   ⇨ `fbPowerMonitor.CallHighPrio();`
7. Optionally, use the following functions—cyclically during operation—to add performance values from sources that are not linked (e.g. in the *CallHighPrio* task):
   ⇨ `fbPowerMonitor.AddPowerFromGrid(rPowerIn:=rPowerSupply);`
   ⇨ `fbPowerMonitor.AddPowerFromLoad(rPowerIn:=rForeignAx-isPwr);`
   ⇨ `fbPowerMonitor.AddPowerFromStorage(rPowerIn:=rStorage-Power);`

⇨ The IEC library can be used.

PowerMonitor

Variables for outputting power values.

Variable name	Description
rActLoadPower	Data type – REAL
	Power values of loads (filtered) [W]
rActLoadPowerRaw	Data type – REAL
	Power values of loads (unfiltered) [W]
rActStoragePower	Data type – REAL
	Power [W] exchanged with the energy storage unit or the devices
rDCVoltage	Data type – REAL
	DC link voltage [V]

Variable name	Description
rPowerStock	Data type – REAL
	Total power flowing in or out of the DC link node [W]. This includes measurement inaccuracies, intrinsic losses, non-measured loads, and the power consumed or output by the storage unit. The sign indicates whether energy is charged or absorbed. Power supplied to the DC link has a negative sign, consumption power a positive sign.
uiPD_PowerIndex_Axis	Data type: UINT
	Start index of the process data word for the power of the axes. Counting starts with index 0. (All axes must use the same 2 process data words for this)

LinkMuMoAxis

Variables for linking SEW MultiMotion axes.

Variable name	Description
itfAxisProcessData	Data type – SEW_IDH.IProcessDataDirectAccess (LogicalDevice of the axis)
	Linking to the process data of the axis
uiPD_PowerIndex	Data type: UINT
	Start index in the process data (index of LowWord and power). Counting starts with index 0.
uiPriorityTask_CycleTime	Data type: UINT
	Bus cycle time in ms

LinkEnergyDevice

Variable for linking MDP92A or MDE90A axes.

Variable name	Description
itfDevice	Data type – SEW_PES_IEnHubCom.IBasicFct (EnergyHub)
	Linking with the energy node
xIsSupply	Data type – BOOL
	• TRUE – Connect supply module • FALSE – Connect storage unit connection

26624737/EN – 03/2020

7 Process data assignment

7.1 Fieldbus connection

If the fieldbus connection is "activated" (→ 🗎 22) in the configuration, a corresponding function block is added to the project when "generating the IEC project" (→ 🗎 23). Furthermore, method calls are generated for interconnection with the FieldbusHandler. As a result, the connection to the higher-level fieldbus runs automatically and no further manual adaptation is required.

The software modules require 8 process data words each for setpoints from a higher-level controller, and they send 8 process data words each via fieldbus.

7.2 Process output data

The following table shows the process output data from the higher-level controller to the inverter for control via fieldbus with 8 process data words.

Word		Bit	Function
PO 1	Control word	0	Enable/emergency stop
		1	Reserved for "enable 2"
		2	Reserved
		3	Reserved
		4	Reserved
		5	Reserved
		6	Reserved
		7	Start/stop
		8	Fault reset
		9	Reserved
		10	• "1": Control B-side • "0": Control A-side (applies only to the MDE90A device)
		11	Reserved
		12	Reserved
		13	Reserved for controller inhibit
		14	Activate standby mode
		15	MOVIKIT® Handshake In

26624737/EN – 03/2020

Word		Bit	Function
PO 2	Setpoint state of charge		• Operating mode 500/ 501 – StateOfChargeControl: Depending on the specified setpoint type: – Absolute voltage value [10^{-1} V] – Percentage value of work envelope [10^{-1}%] – Absolute energy value [kWs] • Operating mode 510 - StorageDischarger: – Absolute voltage value [10^{-1} V]
PO 4	Setpoint Maximum charging power		• Operating mode 500/ 501 – StateOfChargeControl: Depending on the specified setpoint type: – Absolute current [10^{-1} A] – Power limit [10^{1} W].
PO 5	Digital outputs	0	DO 00
		1	DO 01
		2	DO 02
		3	DO 03 (only available with MDP92A)
		4	Relay output
PO 6	Target application mode		Operating mode selection: • 0 – No operating mode activated The device remains in output stage inhibit state (FCB 51) • 500 – StateOfChargeControl • 501 - StateOfChargeControl with active boost mode • 510 – StorageDischarger • 521 - DC synchronization: Disconnect • 522 - DC synchronization: Synchronization and switching on

7.3 Process input data

The following table shows the process input data from the inverter to the higher-level controller for control via fieldbus with 8 process data words.

Word		Bit	Function
PI 1	Status word	0	"1": Ready for operation
		1	Reserved
		2	"1": Output stage enable
		3	"1": Power supply OK All line phases are available and the voltage on all line phases is sufficient for normal operation.
		4	"1": Supply active (B -> A) The device feeds a current ≥ 1 A into the DC link and the output stage of the device is enabled.
		5	"1": Supply active (A -> B) The device feeds a current > 1 A into the connected energy storage unit and the output stage of the device is enabled.
		6	Reserved
		7	"1": Setpoint reached – The active operating mode signals that the setpoint has been reached. • 500/501 – StateOfChargeControl: Setpoint for state of charge reached • 510 – StorageDischarger: Target voltage of the discharge has been reached • 521 - DC synchronization: Storage unit successfully disconnected • 522 - DC synchronization: Storage unit successfully synchronized and switched on
		8	"1": Fault
		9	"1": Warning
		10	• "1": The B-side is the controlled side • "0": The A-side is the controlled side (applies only to the MDE90A device)
		11	"1": State of charge low (warning) The state of charge has a value below the defined operating range.

Word		Bit	Function
		12	"1": State of charge high (warning) The state of charge reached a level above the specified operating range.
		13	"1": Current limit active A current limit is set by the software module and restricts the power supply. **INFORMATION:** The operation of connected axes should be set.
		14	"1": Standby mode active
		15	MOVIKIT® Handshake Out
PI 2	Actual value state of charge		• Operating mode 500/ 501 – StateOfChargeControl: Depending on the specified setpoint type: – Absolute voltage value [10^{-1} V] – Percentage value of work envelope [10^{-1}%] – Absolute energy value [kWs] • Operating mode 510 - StorageDischarger: – Absolute voltage value [10^{-1} V]
PI 3	Status Fault subfault		• With active device fault: Bit 0..7 ErrorSubID, bit 8..15 ErrorID • With active software module error: Error ID of the software module • Without faults/errors or warnings: Active FCB of the connected device
PI 4	Actual value charging power		• Operating mode 500/ 501 – StateOfChargeControl: Depending on the specified setpoint type: – Absolute current [10^{-1} A] – Power [10^{1} W].
PI 5	Digital inputs	0	DI 00
		1	DI 01
		2	DI 02
		3	DI 03
PI 6	Actual application mode		Active operating mode (see PO 6)
PI 7	Actual value voltage		Actual voltage A-side devices [10^{-1} V]
PI 8	Actual value voltage		Actual voltage B-side devices [10^{-1} V]

8 Fault management

8.1 Fault codes

8.1.1 Device communication

16#9000	The process data channel could not be activated during the initialization of the function block. The device might not be connected to the SBUS^PLUS or is not actively supplied with 24 V.
16#9001	The initialization of internal filters and blocks could not be completed.
16#9002	Certain parameters could not be read from the device during initialization. The SBUS^PLUS might not be able to access the device.
16#9003	The initialization of the function block failed during the initial phase. It is possible that the license manager could not be initialized.
16#9004	The device signals a fault. The exact fault code of the device can be found in the "Inverter” structure.
16#9005	The device signals an active FCB (FunctionControlBlock) that is unknown to the control system. It is possible that the library version and the device firmware version do not match.
16#9006	The connected device is not compatible with the software module.
16#19001	Communication error warning: Process data could not be read from the device.
16#19000	The device signals a warning. You can find the warning text in the "Inverter" structure.

8.1.2 DC synchronization

16#90C0	Idle state timeout: The connected axes do not report their idle state within a specified time period.
16#90C1	The connected device (MDP92A, MDE90A) reports a line fault. Therefore, synchronization of the DC link is currently not possible.
16#90C2	Timeout synchronization: The synchronization of the DC link lasts longer than 60 s.
16#90C3	The external function block of the type *IOConnectorDCSync* has not been linked.
16#90C4	Parameter error: The output stage side to which the energy storage unit has been connected has not been selected in the device.
16#90C5	The load-break switch has been opened during the active synchronization.
16#90C6	The "DC Synchronization" operating mode has been selected although the energy storage unit was parameterized as "non-separable" via MOVISUITE®.

26624737/EN – 03/2020

16#90C7	The operating mode measures that the measuring line at the energy storage unit detects an incorrect voltage. The storage may have been disconnected (fuses tripped) or the measuring line may have a wire break.

8.1.3 Multi-stage fuse

16#9040	Interface linking error: Certain interface links are missing
16#9041	The contactor for bypassing the multi-stage fuse does not switch. Although the contactor receives a closing or opening request, the feedback contacts signal the opposite switching state
16#9042	Timeout contactor: The bypass contactor does not close or open in the foreseen time.
16#9043	Fuse error: One of the multi-stage fuses has tripped, the energy storage unit is separated by one of its poles. For example, there could have been a wire break on the measuring line to the storage unit.

8.1.4 StorageMonitor

16#90E0	The connected energy storage unit has reported a general error
16#90E1	The connected energy storage unit reports overtemperature
16#90E2	The connected energy storage unit reports undertemperature
16#90E3	The connected energy storage unit reports a wire break
16#90E4	The connected energy storage unit reports overvoltage

8.1.5 StorageDischarger

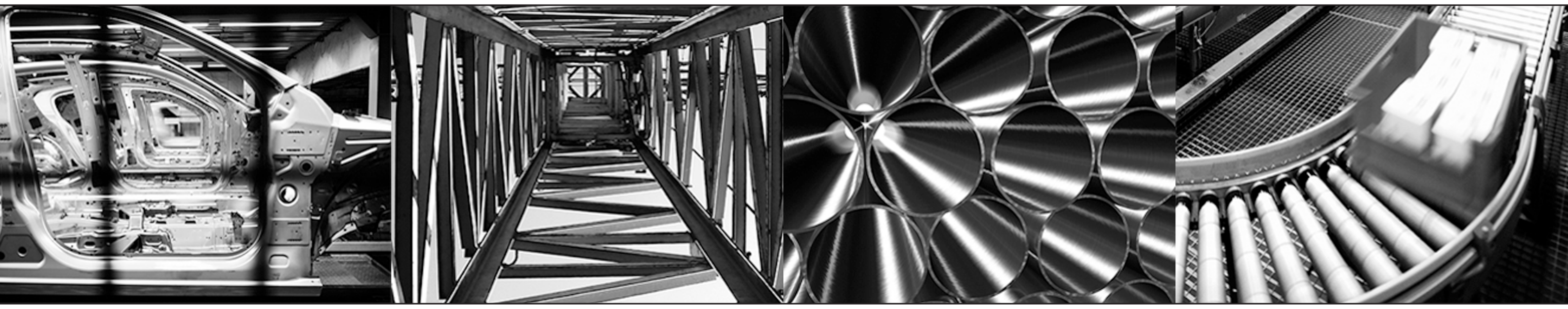
16#9020	Line fault. The MDP92A power supply module signals a grid failure. The discharge function is blocked as discharging is not possible without supply system.
16#9021	External braking resistor fault. The MDP92A power supply module signals that the NC contact of the DI 03 is open. This means that the braking resistor is not mounted correctly or it is already very hot and the bimetal temperature switch has opened.
16#9022	The braking resistor has not been parameterized correctly. The resistance value of the braking resistor is ≤ 0 Ω or the electrical continuous power is ≤ 0 watts. The mode has still been activated.
16#19020	Warning of high power: The discharging module is capable of converting a power that is at least 10% higher than the continuous power of the braking resistor.

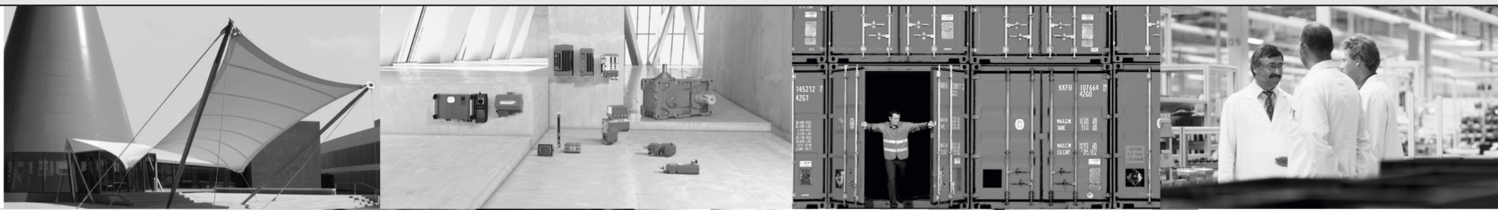
26624737/EN – 03/2020

16#19021	The braking resistor has not been parameterized correctly. The resistance value of the braking resistor is ≤ 0 Ω or the electrical continuous power is ≤ 0 watts.

Index

26624737/EN – 03/20





SEW-EURODRIVE
Driving the world

SEW-EURODRIVE GmbH & Co KG
Ernst-Blickle-Str. 42
76646 BRUCHSAL
GERMANY
Tel. +49 7251 75-0
Fax +49 7251 75-1970
sew@sew-eurodrive.com
→ **www.sew-eurodrive.com**