SEW EURODRIVE

Manual

MOVIKIT® Power and Energy Solutions

DirectMode

Edition 12/2019

29182255/EN

Table of contents

29182255/EN – 12/2019

29182255/EN – 12/2019

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.

29182255/EN – 12/2019

Meaning of the hazard symbols

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

Hazard symbol	Meaning
	General hazard

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

29182255/EN – 12/2019

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 DirectMode	MOVIKIT® DirectMode
MOVIKIT® Power and Energy Solutions DirectMode	Software module
MDP92A power supply module with controlled DC link voltage	MDP92A
DC/DC converter module MDE90A	MDE90A

29182255/EN – 12/2019

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 DirectMode 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.

29182255/EN – 12/2019

3 System description

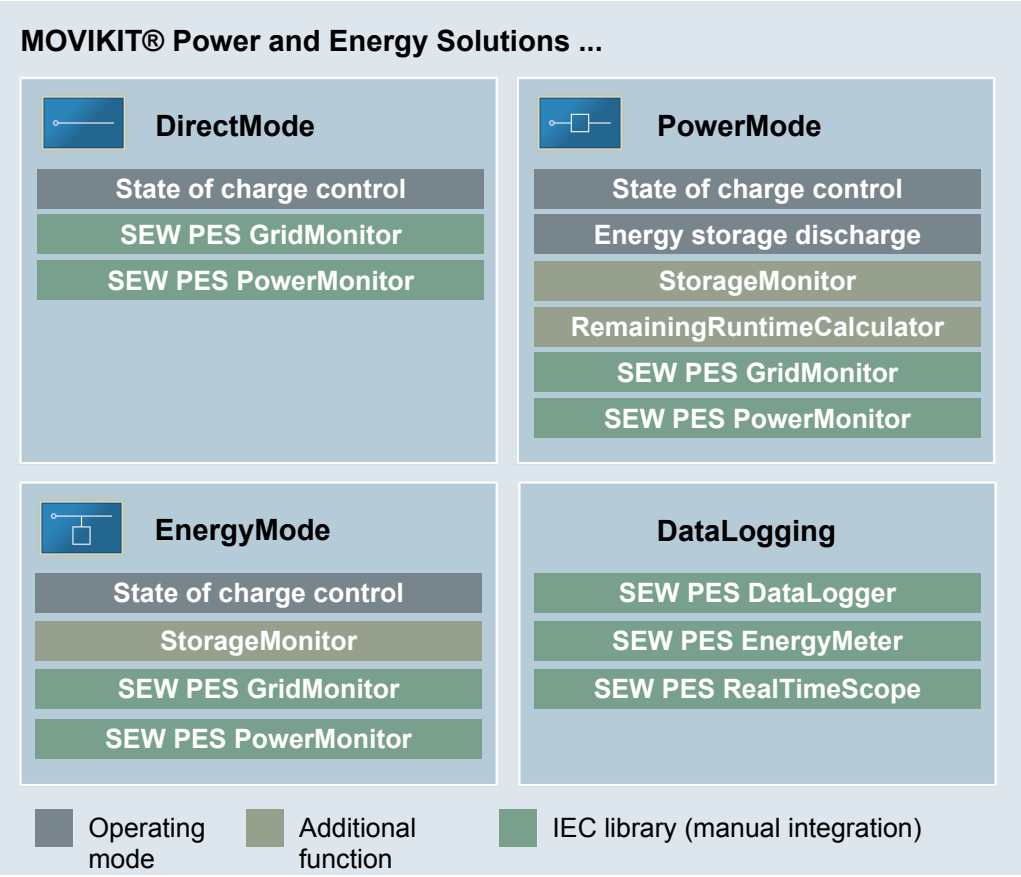
3.1 Module description

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.

Overview

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



31251670155

MOVIKIT® DirectMode

MOVIKIT® DirectMode is primarily designed for applications without energy storage. The software module offers operating modes and IEC libraries for operating the MDP92A and MDE90A devices from the controller, for setting a state of charge in the DC link, and for reading measured values from the devices to use them in the IEC program.

INFORMATION

All minimum required function blocks are integrated by the automatic code generation function.

3.1.1 Operating modes

MOVIKIT® DirectMode provides the following operating modes:

- **State of charge control**

  The state of charge control mode (*StateOfChargeControl*) is used to balance a specific state of charge in the DC link or energy storage by means of a connected device.

For further information, refer to chapter "Operating modes" (→ 🗎 29).

3.1.2 IEC libraries

MOVIKIT® DirectMode 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" (→ 🗎 33).

29182255/EN – 12/2019

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
  - Assigning 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
- **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 AC power systems)
  - Phase and power failure detection and calculation of the remaining system runtime

29182255/EN – 12/2019

4 Project planning information

4.1 Requirement

Correct project planning and proper installation of the devices is 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 for operation:

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

  or

  DC/DC converter module MDE90A

4.3 Software

The following software is required for operation:

- MOVISUITE® engineering software
- MOVIRUN® flexible software platform

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

4.4 Licensing

You need the following license(s) for operation:

- MOVIRUN® flexible

  License for the software platform MOVIRUN® flexible

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).

29182255/EN – 12/2019

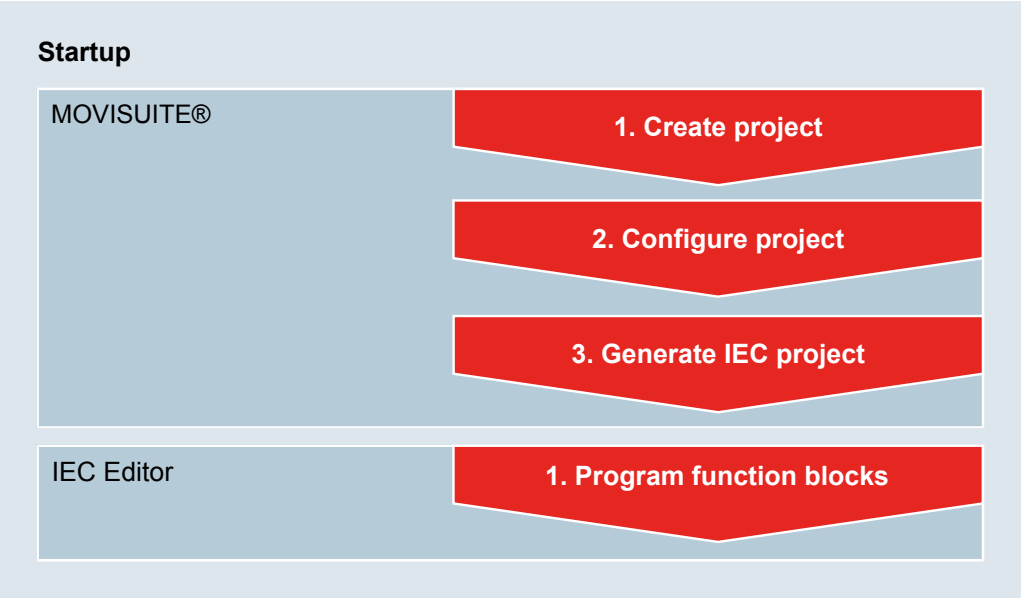
5 Startup

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:



9007227600287243

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.

5.3 Configuring the project

INFORMATION

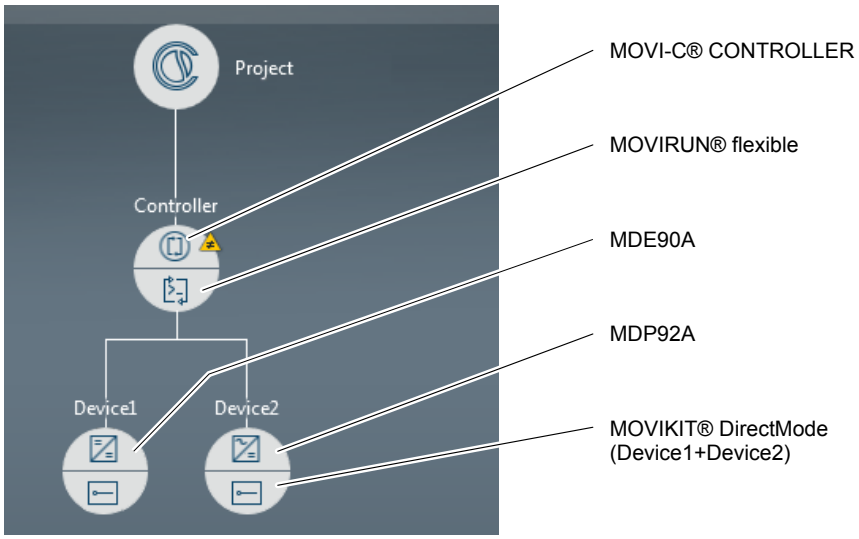
For detailed information on how to operate the MOVISUITE® engineering software, refer to the corresponding documentation.

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

1. Add required device nodes, software nodes (MOVI-C® SoftwareNode) and software modules (see "Sample project") to the project. Devices can be added by performing a network scan or manually by selecting them from the catalog.
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 DirectMode. For detailed information on the configuration of devices or software modules, refer to the respective documentation.

5.3.1 Example project

The following figure shows an example project:



31237315339

5.3.2 Add MOVIKIT® Power and Energy Solutions DirectMode.

INFORMATION

For detailed information on how to operate 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 DirectMode.
   ⇨ A context menu opens.
3. Select the version from the respective drop-down list in the context menu and confirm your selection with [Apply].

⇨ The MOVIKIT® Power and Energy Solutions DirectMode is assigned to the node, the configuration is created, and the basic settings are performed.

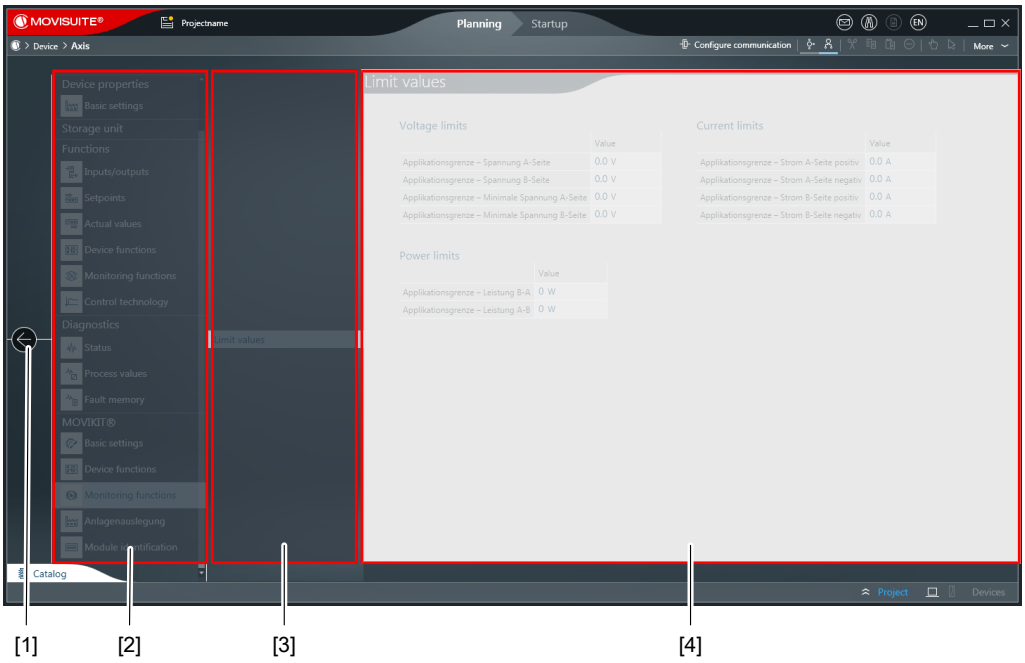
29182255/EN – 12/2019

5.3.3 Configuring MOVIKIT® Power and Energy Solutions DirectMode

INFORMATION

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

1. In MOVISUITE®, click on MOVIKIT® Power and Energy Solutions DirectMode.
   ⇨ The "configuration menus" (→ 🗎 17) of the software module are displayed.



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 for updating the configuration data.

29182255/EN – 12/2019

Configuration menus

The section of the MOVIKIT® Power and Energy Solutions DirectMode in the main menu of the configuration includes the configuration menus described in the following chapters.

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 **Default value: No**

Monitoring functions

Limit values

Configuration of the application limits correspond to the application.

Parameter group	Description
Voltage limits	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- and B-side current • Negative A- 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**

29182255/EN – 12/2019

System design

INFORMATION

As a rule, you have already received the values to be configured during project planning.

Parameter group	Description
Boost charging power	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**

29182255/EN – 12/2019

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.
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** Operation of connected axes is possible 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.
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]

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. Set this value to "8" to use the software module. **Default value: 1** • 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 name	Description
Module identification	Includes name and version for identifying the software module.

29182255/EN – 12/2019

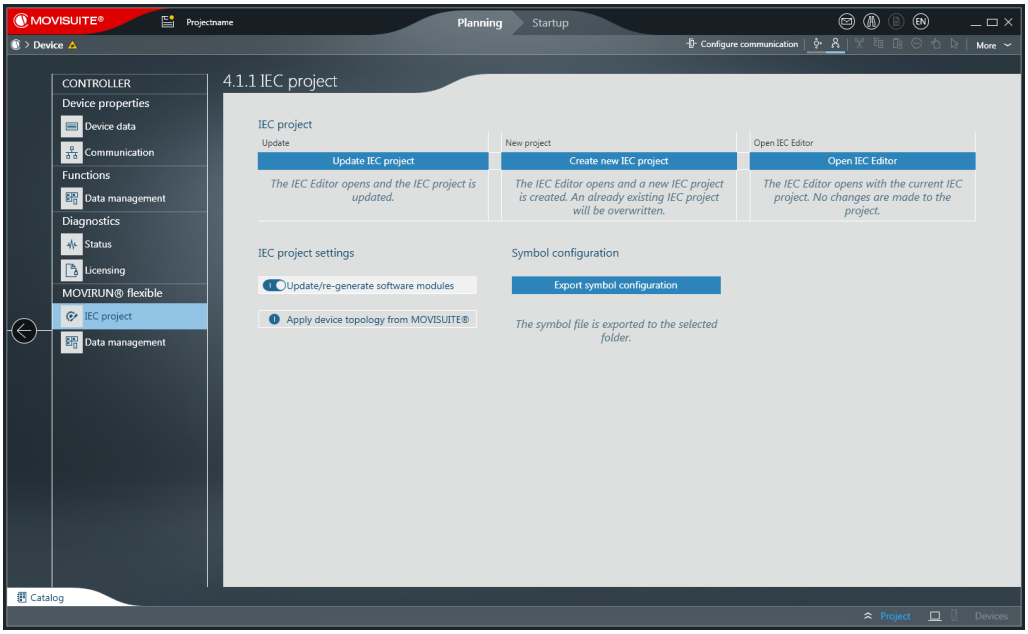
5.4 Generating an IEC project

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

✓ The MOVISUITE® project has been configured.

1. In the function view of MOVISUITE®, click 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].

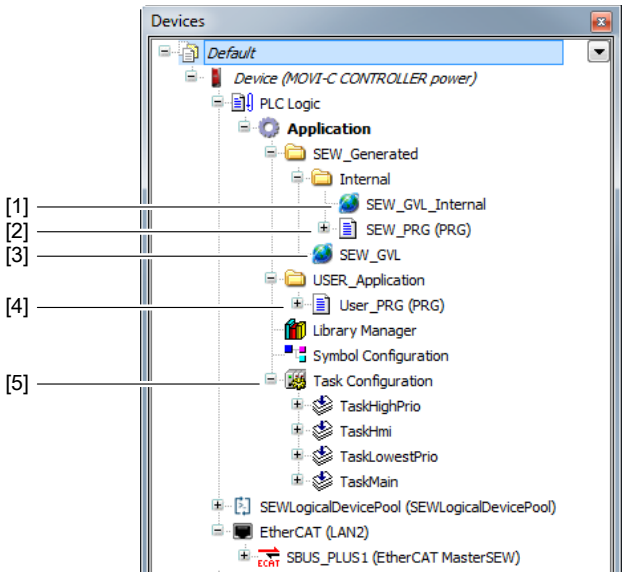
   ⇨ The IEC Editor opens and a new IEC project is created.

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 notification symbol for more information on 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 being generated in time, which means that these axes cannot be operated properly.

29182255/EN – 12/2019

6 IEC programming

6.1 Opening the IEC project

- If an IEC project has already been generated, select the entry [IEC Editor] 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" (→ 🗎 21) chapter.

6.2 Interface in the IEC Editor

The user interface is implemented in the IEC program by an instance in the global variable list *SEW_GVL*. This variable list contains variables for accessing the functions of the software module.

The following diagram shows the interface of the software module in the IEC project:

Interface_MDP92A	SEW_MK_PES_DirectMode.EnergyHubMDP_UI	
xError	BOOL	FALSE
xWarning	BOOL	FALSE
udiMessageID	UDINT	
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	
ModeStateOfChargeControl	SEW_PES_IModeSoCCtrl.ST_ModeSoCControl	

30880259211

29182255/EN – 12/2019

6.3 Basic functions

6.3.1 Diagnostics

Variables for reporting and writing faults and warnings.

Variable name	Description
xError	Data type – BOOL
	• TRUE – Fault present • FALSE – No fault 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.

29182255/EN – 12/2019

6.3.3 Operating mode selection (ModeControl)

The *ModeControl* structure is used to select the "operating mode" (→ 🗎 29) 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)

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]
rActualVoltageL2L3	Data type – REAL
	Current phase-to-phase voltage (RMS) between phases L2 and L3 [V]

29182255/EN – 12/2019

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

29182255/EN – 12/2019

Variable name	Description
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

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 chopped blocked due to fault • FCB_VOLTAGE_CONTROL: 55 Controlling of the DC link voltage activated
wDigitalInputs	Data type: WORD
	State of the digital inputs
usiErrorID	Data type: USINT
	Error ID
usiErrorSubID	Data type: USINT
	Suberror ID

29182255/EN – 12/2019

Variable name	Description
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 example of a code shows how to access measured values via the respective OUT structures to use them, for example, for further 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;
```

29182255/EN – 12/2019

6.4 Operating modes

6.4.1 State of charge control (StateOfChargeControl)

The state of charge control operating mode *(StateOfChargeControl)* is used to balance a specific state of charge in the DC link or in the energy storage unit by means of the connected device (such as MDP92A, MDE90A).

INFORMATION

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

INFORMATION

The operating type automatically limits the setpoint for the state of charge to the specified "operating range" (→ 🗎 30). Additionally, the setpoints for the charge limit are automatically limited in such a way that the "application limits" (→ 🗎 17) 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" (→ 🗎 19):

- As absolute voltage in [V] – Adjust the state of charge according to the specified absolute DC link voltage.
- As percentage value (between 0% and 100%) – Adjust the state of charge depending on the operating range specified in the "configuration" (→ 🗎 17).
- 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:

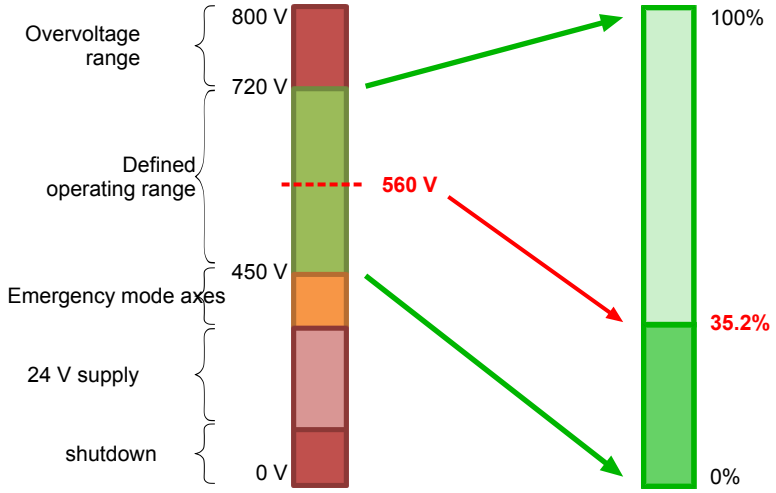
- As power limit – Limit the device current on the respective control side so that the specified power limit results with the current voltage.
- As current limit – Limit the maximum device current on the respective control side to the specified value.

29182255/EN – 12/2019

Specifying the operating range

A valid operating range must be specified in the "Configuration" (→ 🗎 17) to being able to use the operating mode. Setpoints to the device for the state of charge are always limited to this operating range irrespective of the unit set. This can be a range between 450 V and 720 V, for example (see figure). This means 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. 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 voltage value is included quadratically in the calculation of the amount of energy in the DC link 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.

29182255/EN – 12/2019

IN

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

Variable name	Description
xStart	Data type: BOOL
	• TRUE – Start • FALSE – Stop
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]

29182255/EN – 12/2019

Variable name	Description
eOperatingRange	Data type – ENUMERATION
	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, if necessary even in reduced operation. • BELOW_24V_SUPPLY The state of charge is too low for operating axes. However, 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.

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 example of a code shows how the voltage of the DC link on the MD-P92A power supply module is set to 650 V and a maximum charging power of 40 kW is set.

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;
```

29182255/EN – 12/2019

6.5 IEC libraries

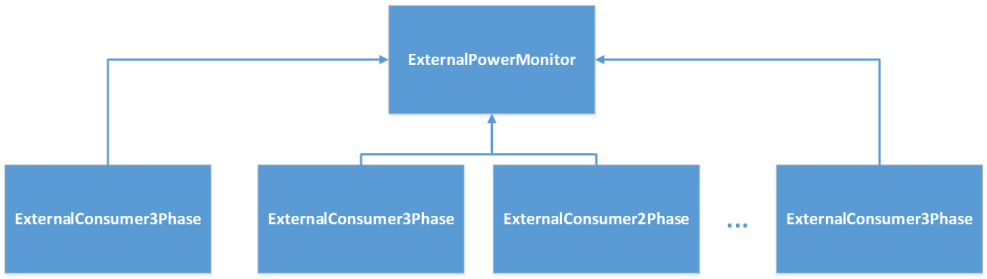
6.5.1 SEW PES GridMonitor

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 graphic illustrates this functional principle:

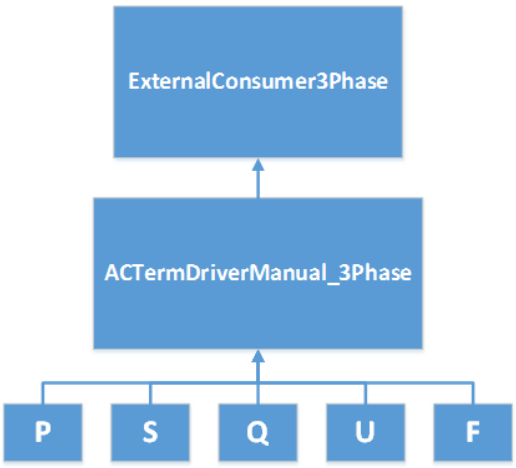


28625588875

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 "ExternalPowerMonitor" function block.
rActivePowerSum	Data type – REAL
	Total active power of all connected loads in [W]

29182255/EN – 12/2019

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

29182255/EN – 12/2019

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]

29182255/EN – 12/2019

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.

29182255/EN – 12/2019

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);
   ```

29182255/EN – 12/2019

⇨ 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 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.ExternalPowerMon-
   itor;
   ```

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.

6.5.2 SEW PES PowerMonitor

INFORMATION

The function block is not integrated via automatic code generation but must be implemented manually in the project.

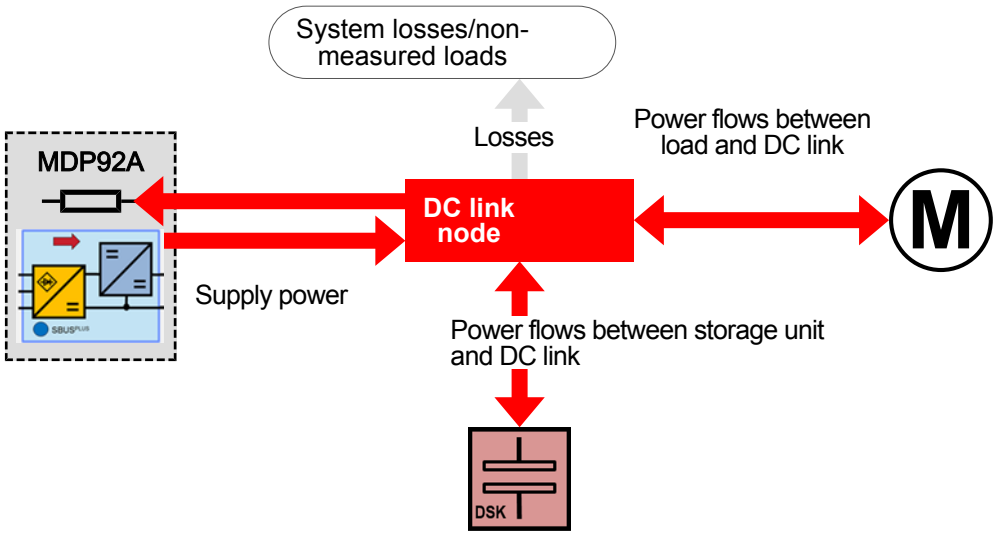
The *SEW PES PowerMonitor* IEC library systematically groups incoming and outgoing instantaneous power values.

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, the process data profile must leave at least 2 process data words for the actual values (ideally the last two. If the last two process data words are not used, another step is necessary during initialization.) For the 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]. Parameter 8364.212 is then inserted into the process data for the actual values (low word and high word).

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

29182255/EN – 12/2019

2. Generate an IEC project based on this configuration. See chapter "Generating an IEC project" (→ 🗎 21).
3. Create an instance of the *PowerMonitor* function block.
   ⇨ `fbPowerMonitor:SEW_MK_PES_DirectMode.SEW_PwrMon.PowerMonitor;`
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, //Leistung wurde auf PE15 und PE16 gemappt
uiPriorityTask_CycleTime:=gc_uiTaskCycleTime);
```
6. Call up the following function in the *CallHighPrio* task:
   ⇨ `fbPowerMonitor.CallHighPrio();`
7. Alternatively, use the following functions to add power values from sources that are not linked:
   ⇨ `fbPowerMonitor.AddPowerFromGrid(rPowerIn:=rPowerSupply);`
   ⇨ `fbPowerMonitor.AddPowerFromLoad(rPowerIn:=rForeignAxisPwr);`
   ⇨ `fbPowerMonitor.AddPowerFromStorage(rPowerIn:=rStoragePower);`

⇨ 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]

29182255/EN – 12/2019

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

29182255/EN – 12/2019

7 Process data assignment

7.1 Fieldbus connection

If the fieldbus connection is "activated" (→ 🗎 20) in the configuration, a corresponding function block is added to the project when "generating the IEC project" (→ 🗎 21). 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
PO 2	Setpoint state of charge (operating mode 500 StateOfChargeControl)		Depending on the specified setpoint type: • Absolute voltage value [10^{-1} V] • Percentage value of work envelope [10^{-1}%] • Absolute energy value [kWs]

29182255/EN – 12/2019

Word		Bit	Function
PO 4	Setpoint Maximum charging capacity (operating mode 500 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
		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)

29182255/EN – 12/2019

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 for state of charge reached (operating mode 500 – StateOfChargeControl)
		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.
		12	"1": State of charge high (warning) The state of charge reached a level above the specified operating range.
		13	"1": Reserved
		14	"1": Standby mode active
		15	MOVIKIT® Handshake Out
PI 2	Actual value state of charge (operating mode 500 StateOfChargeControl)		Depending on the specified setpoint type: • Absolute voltage value [10^{-1} V] • Percentage value of work envelope [10^{-1}%] • Absolute energy value [kWs]

29182255/EN – 12/2019

Word		Bit	Function
PI 3	State 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 capacity (operating mode 500 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]

29182255/EN – 12/2019

8 Error management

8.1 Error codes

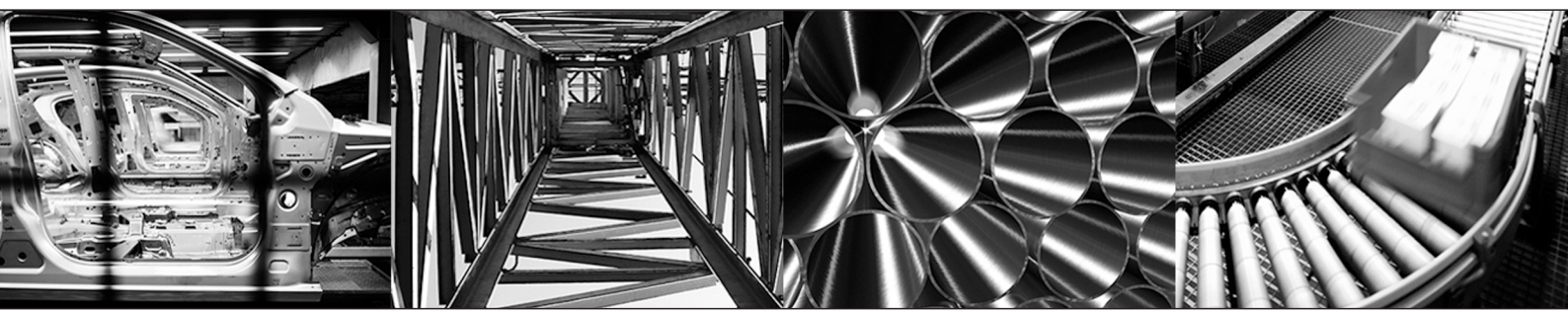
8.1.1 Device communication

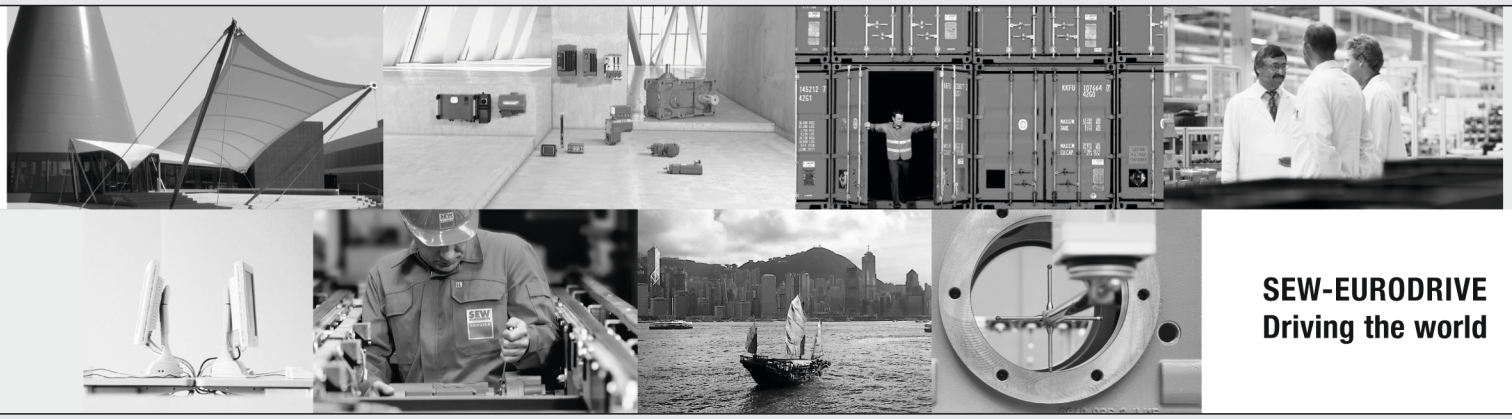
udiMessageID	Description
16#9000	The process data channel could not be activated during the initialization of the function block. The device might not be connected to the SBUS^{PLUS} or is not actively supplied with 24 V.
16#9001	The initialization of internal filters and blocks could not be completed.
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 find the warning text in the "Inverter" structure.

29182255/EN – 12/2019

Index

29182255/EN – 12/19





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**