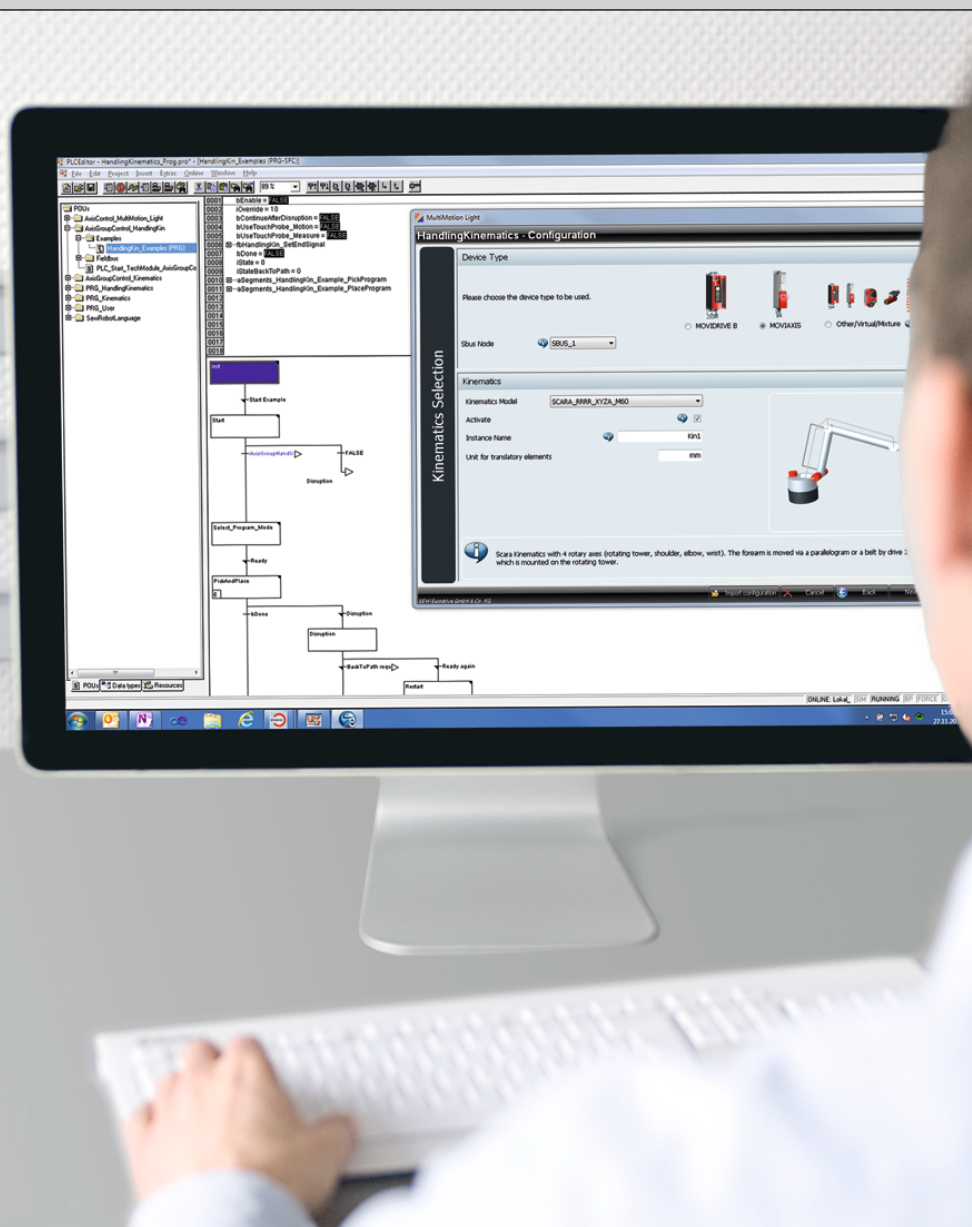




Manual



Technology Module **HandlingKinematics for MultiMotion/MultiMotion Light**



Contents

1	General information.....	5
1.1	About this documentation	5
1.2	Structure of the safety notes	5
1.2.1	Meaning of signal words	5
1.2.2	Structure of section-related safety notes.....	5
1.2.3	Structure of embedded safety notes	6
1.3	Right to claim under warranty	6
1.4	Exclusion of liability	6
1.5	Applicable documentation.....	7
1.6	Product names and trademarks.....	7
1.7	Copyright notice	7
2	Safety notes	8
2.1	General	8
2.2	Target group	8
2.3	Designated use	9
2.4	Bus systems.....	9
2.5	Short designation	10
3	Project planning notes	11
3.1	Requirements.....	11
3.2	Hardware	11
3.3	Software	11
3.4	Technology points.....	11
4	Technology description	14
4.1	Areas of application	14
4.2	Operating modes	14
4.3	Functions	15
4.4	Tools for configuration and diagnostics	16
4.5	Useful contents in other manuals.....	17
4.5.1	CCU HandlingKinematics application module.....	17
4.5.2	Kinematics for MultiMotion/MultiMotion Light technology module.....	17
5	Startup	18
5.1	Requirements.....	18
5.2	MOVITOOLS® MotionStudio engineering software	18
5.3	Startup procedure	19
5.4	Step 1: Creating a MotionStudio® project	20
5.5	Step 2: Creating a PLC Editor project.....	22
5.6	Step 3: Integrating the HandlingKinematics technology module.....	23
5.6.1	Manually integrating the HandlingKinematics technology module	24
5.7	Step 4: Copying SRL files to the controller memory card	26
5.8	Step 5: Starting up the single axes	28
5.9	Step 6: Configuring the kinematic model	29
5.9.1	Adding the HandlingKinematics technology module	29
5.9.2	Configuring the "HandlingKinematics" technology module	31

6	MOVI-PLC® program	32
6.1	Task configuration.....	32
6.2	Control via fieldbus	33
6.2.1	Activating the fieldbus mapping	33
6.2.2	Using the process data monitor	34
6.3	Example for a sequential program in the controller	36
6.4	Functions for application programming.....	39
6.4.1	MC_HandlingKin_GetActWaitSignal	39
6.4.2	MC_HandlingKin_ResetAllWaitSignals	40
6.4.3	MC_HandlingKin_ResetWaitSignal.....	40
6.4.4	MC_HandlingKin_SetEndSignal_FUN	40
7	Variable interfaces	42
7.1	Variable interface AxisGroupHandlingKin.Inst[.]	42
7.1.1	Control variables: AxisGroupHandlingKin.Inst[.].In	43
7.1.2	Status variables: AxisGroupHandlingKin.Inst[.].Out	47
7.2	Variable interface AxisGroupHandlingKinProfile.Inst[.].....	49
7.2.1	Control variables: AxisGroupHandlingKinProfile.Inst[.].Standard.In	50
7.2.2	Status variables: AxisGroupHandlingKinProfile.Inst[.].Standard.Out	52
8	Applications	58
8.1	Controlling the B/C degree of freedom and the A7/A8 auxiliary axes.....	58
8.2	Changing the existing fieldbus mapping or integrating a user fieldbus mapping	60
8.2.1	Fieldbus mapping programs.....	61
8.2.2	Flow diagrams of fieldbus mapping programs	62
8.2.3	Adjusting the templates in the process data monitor	65
8.3	Integrating a customized kinematics model	65
8.4	Simultaneous use of Kinematics and HandlingKinematics	65
	Index	66

1 General information

1.1 About this documentation

The documentation is part of the product and contains important information. The documentation is for everyone who works with this product.

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

1.2 Structure of the safety notes

1.2.1 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 drive system or its environment.
INFORMATION	Useful information or tip: Simplifies handling of the drive system.	

1.2.2 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.

Meaning of the hazard symbols

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

Hazard symbol	Meaning
	General hazard

Hazard symbol	Meaning
	Warning of dangerous electrical voltage
	Warning of hot surfaces
	Warning of risk of crushing
	Warning of suspended load
	Warning of automatic restart

1.2.3 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.

1.3 Right to claim under warranty

A requirement of fault-free operation and fulfillment of any rights to claim under limited warranty is that you adhere to the information in the documentation at hand. Therefore, read the documentation before you start working with the software and the connected units from SEW-EURODRIVE.

Make sure that the documentation is available to persons responsible for the machinery and its operation as well as to persons who work independently on the units. Also ensure that the documentation is legible.

1.4 Exclusion of liability

Please observe this documentation as well as the documentation for the software used and the SEW-EURODRIVE devices connected. This documentation must be observed to ensure that the devices operate safely and that the specified product properties and performance characteristics are achieved.

SEW-EURODRIVE assumes no liability for injury to persons or damage to equipment or property resulting from non-observance of the documentation. In such cases, SEW-EURODRIVE assumes no liability for defects.

1.5 Applicable documentation

Observe the following applicable documents:

- **"Technology Module Kinematics for MultiMotion/MultiMotion Light" manual**
- **"HandlingKinematics CCU Application Module" manual**
- "MOVIAXIS® MX Multi-Axis Servo Inverter" operating instructions
- "MOVIDRIVE® MDX Drive Inverters" operating instructions
- "DH.21B (standard) / DH.41B (advanced) Controller" manual
- "UHX71B (power) Controller" manual
- "MultiMotion Program Module, Universal, Parameterizable Software Platform for MOVI-PLC®" manual
- "MOVI-PLC® Programming in the PLC Editor" system manual

Always use the latest edition of documentation and software.

The SEW-EURODRIVE homepage (www.sew-eurodrive.com) provides a wide selection of documentation in various languages available for download.

If required, you can also order printed and bound copies of the documentation from SEW-EURODRIVE.

1.6 Product names and trademarks

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

1.7 Copyright notice

© 2015 SEW-EURODRIVE. All rights reserved.

Unauthorized reproduction, modification, distribution or any other use of the whole or any part of this documentation is strictly prohibited.

2 Safety notes

2.1 General

The following basic safety notes are intended to prevent injury to persons and damage to property. The user must ensure that the basic safety notes are read and observed.

Ensure that persons responsible for the machinery and its operation as well as persons who work 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, please contact SEW-EURODRIVE.

The following safety notes refer to the use of the software. Also observe the supplementary safety notes in this documentation and in the documentation for the connected units from SEW-EURODRIVE.

This document does not replace the detailed documentation for the connected units. This documentation assumes that the user has access to and is familiar with the documentation for all connected units from SEW-EURODRIVE.

Never install or operate damaged products. Report any damage to the shipping company immediately.

Depending on the degree of protection, units may have live, uninsulated, and sometimes moving or rotating parts, as well as hot surfaces during operation.

Removing required covers without authorization, improper use or incorrect installation and operation may result in severe injury to persons, or damage to machinery. Consult the documentation for further information.

2.2 Target group

Work with the software in this solution may only be performed by adequately qualified personnel. Qualified personnel in this context are persons who have the following qualifications:

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

All mechanical work on connected units is to be performed exclusively by adequately qualified personnel. Qualified personnel in the context of this documentation are persons familiar with the design, mechanical installation, troubleshooting and servicing of the product, who possess the following qualifications:

- Training in mechanical engineering, e.g. as a mechanic or mechatronics technician (final examinations must have been passed).
- Knowledge of this documentation and other applicable documentation.

All electrical work on connected units is to be performed exclusively by adequately qualified electricians. Qualified electricians in the context of this documentation are persons familiar with electrical installation, startup, troubleshooting and servicing of the product, who possess the following qualifications:

- Training in electrical engineering, e.g. as an electrician or mechatronics technician (final examinations must have been passed).
- Knowledge of this documentation and other applicable documentation.

- Knowledge of the relevant safety regulations and laws.
- Knowledge of all other standards, directives and laws named in this documentation.

The above-mentioned persons must have the express authorization of the company to operate, program, configure, label and ground units, systems and circuits in accordance with the standards of safety technology.

All work in the areas of transportation, storage, operation and waste disposal must be carried out by persons who are trained appropriately.

2.3 Designated use

The "MultiMotion/MultiMotion Light" program module with the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module is a universal, parameterizable software platform for the controllers in the freely programmable designs MOVI-PLC® advanced and power by SEW-EURODRIVE.

The "MultiMotion/MultiMotion Light" program module consists of the following components:

- **Project template**
 - `AxisControl_MultiMotion.pro` for the program variant *MultiMotion*
 - `AxisControl_MultiMotion_Light.pro` for the program variant *MultiMotion Light*

The user creates a PLC Editor project from the project template. Then, the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module is integrated with the suitable task configuration. This fulfills all the prerequisites to be able to access the kinematics functions in the MOVI-PLC® program.

- **MultiMotion Editor**

The MultiMotion Editor serves for parameterization of the functions in the MOVI-PLC® program, as well as for diagnostics and testing the parameterized functions.

The "MultiMotion/MultiMotion Light" program module with the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module is not a complete application solution but rather a software template to which users have to add their own program parts. Without these additional program parts, the "MultiMotion/MultiMotion Light" program module is not suited for controlling automatic processes in machines.

Test in control mode

When testing of the parameterized functions with the MultiMotion Editor, you can directly access the variable interface in control mode, and thus the MotionControl functions as well. The variable interface is no longer controlled by the program parts added by the user, which means that limits and inhibits in the user program might no longer be effective. Suitable precautionary measures must therefore be taken when using control mode. Using control mode is the sole responsibility of the user.

2.4 Bus systems

A bus system makes it possible to adapt frequency inverters and/or motor starters to the particulars of the machinery within wide limits. This results in the risk that a change of parameters that cannot be detected externally can result in unexpected, though not uncontrolled, system behavior.

2.5 Short designation

The following short designations are used in this documentation.

Type designation	Short designation
"HandlingKinematics for MultiMotion/MultiMotion Light" technology module	HandlingKinematics for MultiMotion/MultiMotion Light technology module
"MultiMotion/MultiMotion Light" program module	Program module
Program variant <i>MultiMotion</i>	<i>MultiMotion</i>
Program variant <i>MultiMotion Light</i>	<i>MultiMotion Light</i>
MultiMotion Editor in 2 versions <i>MultiMotion</i> and <i>MultiMotion Light</i>	MultiMotion Editor
Controller in freely programmable MOVI-PLC® design	Controller, Controller MOVI-PLC®
Higher-level controller (<i>Programmable Logic Controller</i>)	PLC

3 Project planning notes

3.1 Requirements

Correct configuration and proper installation of the units are required for successfully starting up and operating the application modules with the Application Configurator.

You find detailed configuration information in the documentation of the respective units (see chapter "Other applicable documentation").

3.2 Hardware

Using the technology module requires the following hardware:

- Controller in freely programmable MOVI-PLC® design:
 - MOVI-PLC® advanced: DHE41B/DHF41B/DHR41B
1 kinematic instance per controller can be operated, meaning 1 kinematic model can be controlled.
 - MOVI-PLC® power: UHX71B
3 kinematic instances per controller can be operated, meaning 3 kinematic models can be controlled. For each of these 3 kinematic instances can be selected individually, if they should use the "Kinematics" technology module or "HandlingKinematics for MultiMotion/MultiMotion Light".
- SD memory card OMC41B or CFast memory card OMH71B with sufficient technology points. The required number of technology points depends on the desired functionality, see chapter "Technology points" (→ 11).

3.3 Software

Using the technology module requires the following software:

- MOVITOOLS® MotionStudio 6.1 engineering software (service pack 1)
- "MultiMotion/MultiMotion Light" program module (version V160R100 or higher)
- "Kinematics technology module" Kinematics software package (version V160R100 or higher)

The complete software required is automatically included in the installation of MOVITOOLS® MotionStudio 6.1 (service pack 1) or higher.

You can download the software from the SEW-EURODRIVE homepage (www.sew-eurodrive.com).

3.4 Technology points

A certain amount of technology points on the controller memory card is required to use the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module. Generally, 5 technology points per kinematic instance are required for HandlingKinematics. When the kinematic model CARTESIAN GANTRY is used, 4 technology points per kinematic instance are sufficient.

The following table shows, how many technology points are used per kinematic instance of the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module.

Kinematic models	Technology points per kinematics instance
CARTESIAN GANTRY	4
Other kinematic model	5

In addition to the technology points for the use of kinematic instances, further technology points are required for the following functions. They are only required once per controller, not per kinematic instance.

Function	Additional technology points per controller
3D simulation (see following note)	+10
Program module in program variant <i>MultiMotion</i> (see the following note "Program variant <i>MultiMotion</i> or <i>MultiMotion Light</i> ")	+2

INFORMATION



Using the 3D simulations during startup

To adjust the kinematic model to the actual robot, and to simulate and test the motion path, SEW-EURODRIVE recommends **always using the 3D simulation for checking the configuration and for programming.**

- The 3D simulation is not possible when the simulation PC is connected via USB or fieldbus. The 3D simulation only communicates via Ethernet engineering interface (X37).
- For this purpose, establish an Ethernet connection between controller and simulation PC.

INFORMATION



Full version of the 3D simulation

Usually the full version is only required on the controller that is used for simulating and testing the motion and sequential program of the PLC in the office.

INFORMATION



Program variant *MultiMotion* or *MultiMotion Light*

The program variant *MultiMotion* is NOT required for the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module. The program variant *MultiMotion Light* is sufficient for the kinematic control.

The program variant *MultiMotion* is only required when other MultiMotion axes must be inserted besides the kinematic model, and the program variant *MultiMotion Light* is not sufficient for those axes.

The technology module consumes the technology points required for the kinematic instances directly at controller startup. The 10 further technology points for the 3D simulation are only used 30 minutes after controller startup. During this time, the 3D simulation can be used without additional 10 technology points. When not enough technology points are available, the 3D simulation is deactivated after the test time of 30 minutes.

4 Technology description

The "HandlingKinematics for MultiMotion/MultiMotion Light" technology module provides the functionality of the CCU application module "HandlingKinematics" in the programmable MultiMotion environment on the MOVI-PLC® controller.

4.1 Areas of application

Use without higher-level controller (PLC)

Unlike the "HandlingKinematics" CCU application module, the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module does not require a higher-level controller to control via fieldbus. It is possible to freely program in PLC Editor. Using this possibility, you can create the sequential program on the MOVI-PLC® controller according to IEC61131.

Use with higher-level controller (PLC)

The "HandlingKinematics for MultiMotion/MultiMotion Light" technology module can be operated at a higher-level controller, just like the "HandlingKinematics" CCU application module. The fieldbus interface of the application module is available and can be easily used. Furthermore, the fieldbus interface can be adjusted to your requirements. You can freely adjust, which tasks should be adopted by the higher-level controller and which tasks should be performed in addition to the functions of the HandlingKinematics on the controller MOVI-PLC®.

4.2 Operating modes

The "HandlingKinematics for MultiMotion/MultiMotion Light" technology module offers the following operating modes:

- Jog mode axis by axis *Jog Axis* (**HANDLING_KIN_JOG_AXIS**)
Jogging of the kinematic model in the axis coordinate system
- Cartesian jog mode *Jog Cartesian* (**HANDLING_KIN_JOG_CART**)
Jogging of the kinematic model in the Cartesian translation and orientation coordinates
- Referencing mode *Homing* (**HANDLING_KIN_HOMING**)
Referencing the axes
- Program mode *Program Auto* (**HANDLING_KIN_PROGRAM_AUTO**)
Executing programs in automatic mode, without confirmation of each program step. Here, 4 SRL¹⁾ programs are provided, that differ in the way, of how the path points are hit (various interpolation data):
 - Program 1: TARGET AXIS (SRL_TargetAxis)
The programmed poses are hit with axis interpolation (kinematics mode KIN_TARGET_AXIS).
 - Program 2: TARGET CART (SRL_TargetCart)
The programmed poses are hit with Cartesian interpolation (kinematics mode KIN_TARGET_CART).
 - Program 3: LINEAR coordinated (SRL_Lin3DCoordinated)

The programmed poses are hit with path interpolation (kinematics mode KIN_LIN_3D). The orientation guide is performed coordinated.

- Program 4: LINEAR synchronized (SRL_Lin3DSynchronized)

The programmed poses are hit with path interpolation (kinematics mode KIN_LIN_3D). The orientation guide is performed synchronous to the relevant path segment.

- Program mode *Program Step* (**HANDLING_KIN_PROGRAM_STEP**)

Executing programs in step mode, meaning with confirmation of each program step. This operating mode is used for diagnostics of the automatic mode. The programs can be processed step-by-step, meaning SRL-program line by program line.

For detailed information on the operating mode, refer to the "HandlingKinematics CCU application module" manual.

1) SRL = SEW Robot Language

4.3 Functions

The "HandlingKinematics for MultiMotion/MultiMotion Light" technology module offers the following functions:

- **Functions of the "MultiMotion/MultiMotion Light" program module**

For a detailed description of all usable functions of the "MultiMotion/MultiMotion Light" program module, refer to the manual "MultiMotion Program Module: Universal, Parameterizable Software Platform for MOVI-PLC®".

- **Functions of the "HandlingKinematics" CCU application module**

Using the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module, you can use the entire functionality of the "HandlingKinematics" CCU application module, embedded in the programmable software platform MultiMotion.

For detailed information on the functions, refer to the "HandlingKinematics CCU application module" manual.

- **Additional functions compared to the "HandlingKinematics" CCU application module**

- Adjusting the fieldbus interface to certain requirements
- Programming the motion sequence and data storage directly in the controller, for example target pose
- Shorter cycle times by using the MOVI-PLC® power controller
- Using the SBus^{PLUS} system bus
- Using the user kinematic model
- Using the B/C degree of freedom and the A7/A8 auxiliary axes
- Using the user SRL programs
- and others

The "HandlingKinematics for MultiMotion/MultiMotion Light" technology module uses 8 MultiMotion axes per configured kinematic instance.

4.4 Tools for configuration and diagnostics

The following tools are available when using the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module:

- **Kinematics Configurator**

The technology module can be added and parameterized via Kinematics Configurator.

- **HandlingKinematicsMonitor**

The HandlingKinematicsMonitor offers module specific diagnostics information on variable interface, operating states and faults. The actual HandlingKinematics variables are visualized and can be manually controlled here.

- **Process data monitor (PD-Monitor)**

The process data monitor is used for diagnostics, introduction and manual control of the fieldbus interface. The process data monitor displays fieldbus input data and fieldbus output data that are exchanged between the controller and the higher-level controller. The process data monitor only accesses data of the fieldbus interface.

- **MessageHandler**

The MessageHandler is a diagnostics tool that displays information on errors in plain text and in a clear manner.

- **3D simulation**

The 3D simulation allows for automatic simulation of the kinematic model and is used for the following tasks:

- Configuring the kinematic model and adaptation to the actual robot
- Learning how to use the kinematic control system
- Checking motion sequences
- Offline programming

4.5 Useful contents in other manuals

4.5.1 CCU HandlingKinematics application module

For detailed information on the following topics, refer to the manual "CCU HandlingKinematics application module":

- Standardized fieldbus interface:
 - For structure and meaning of the signals, refer to chapter "Communicating and process data exchange" and chapter "Process data assignment"
 - Mechanism for consistency on the fieldbus, see chapter "Consistent data transmission"
- Operating modes and functions
- Flow diagrams and cycle diagrams for control of the technology module
- SRL programs 1 – 4 in program mode
- Kinematic configuration with Kinematics Configurator
- Operating the HandlingKinematicsMonitor
- Error codes
- Glossary and index
- Troubleshooting, especially in case of functional limitations

4.5.2 Kinematics for MultiMotion/MultiMotion Light technology module

For detailed information on the following topics, refer to the manual "Kinematics for MultiMotion/MultiMotion Light technology module":

- Basics of robotics:
 - Kinematic models
 - Interpolating operating modes
 - Coordinate systems
- Helpful information for selecting the following modules:
 - "Kinematics for MultiMotion/MultiMotion Light" technology module
 - "HandlingKinematics for MultiMotion/MultiMotion Light" technology module
 - CCU "HandlingKinematics" application module
- Glossary and index

5 Startup

5.1 Requirements

The requirements listed in chapter "Project planning notes" (→ 11) must be met for a correct startup.

Check the inverter installation, the encoder connection and the controller installation based on the installation notes in the operating instructions and the fieldbus manual.

For installation notes, refer to the documentations of the respective units and software components, as listed in chapter "Applicable documentation" (→ 7).

Special kinematics settings that go beyond the basic functionality require the MotionStudio® authorization level to be changed to "Advanced" or "Advanced configuration."

5.2 MOVITOOLS® MotionStudio engineering software

You need the MOVITOOLS® MotionStudio engineering software for startup. The following tools are included in the delivery:

- **PLC Editor** is a programming environment for SEW-EURODRIVE controllers. PLC Editor allows for creating and debugging of programs according to standard IEC 61131-3.
- **Drive Startup for MOVI-PLC®** is required to set the inverter on the motor (motor startup) and to ensure communication with the controller.
- **"MultiMotion/MultiMotion Light" program module** with the following tools:
 - MultiMotion Editor
 - Kinematics Configurator

INFORMATION

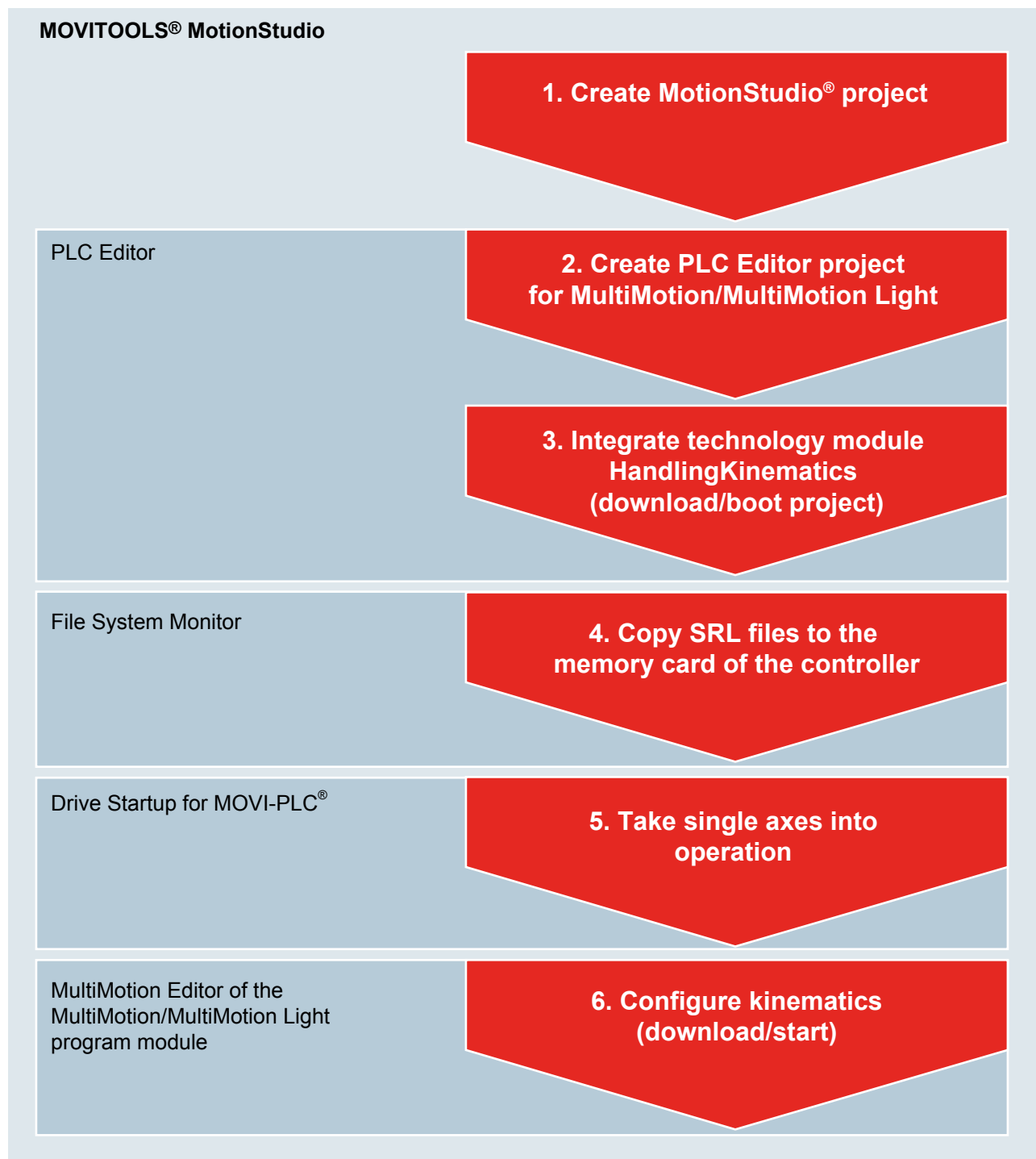


It is strongly recommended, to use the following tools for diagnostics:

- HandlingKinematicsMonitor
 - Module-specific diagnostics information to variable interface, to operating states and faults
 - Visualization and manual control of the HandlingKinematics variables
- 3D simulation
 - Adjustment of configuration and real robot. Only when the data match is the configuration correct.
 - Start via the menu bar in MOVITOOLS® MotionStudio, [Settings] > [Extras], or via the module diagnostics
- MessageHandler
 - Detailed information in plain text, for example error messages, warnings
 - Start via the controller context menu in MOVITOOLS® MotionStudio, [Diagnostics]

5.3 Startup procedure

The following figure shows the entire procedure step-by-step.



15855078027

5.4 Step 1: Creating a MotionStudio® project

Further information on MOVITOOLS® MotionStudio is available in the documentation for this software.

Proceed as follows:

- ✓ The suitable controller with the required firmware version is available.
 - ✓ The required amount of technology points on the controller memory card is available.
1. Connect the engineering PC/laptop with the controller via Ethernet (X37 interface).

INFORMATION



When carrying out engineering via Ethernet, make sure the IP address of the PCs engineering interface and of the connected controller is in the same subnetwork. The default IP address of the controller is 192.168.10.4. This means the IP address of the engineering interface must be set to 192.168.10.xxx.

2. Supply voltage to the controller.
3. Start MOVITOOLS® MotionStudio.
4. Create a new project in MOVITOOLS® MotionStudio in the menu bar via [Project] > [New].
 - ⇒ The project file is saved in the specified target directory.
5. Configure the engineering interface that you want to use for accessing the controller in the menu bar via [Network] > [Communication connections].
6. To find the controller, scan the network in the connection mode "Online" with the symbol [1].

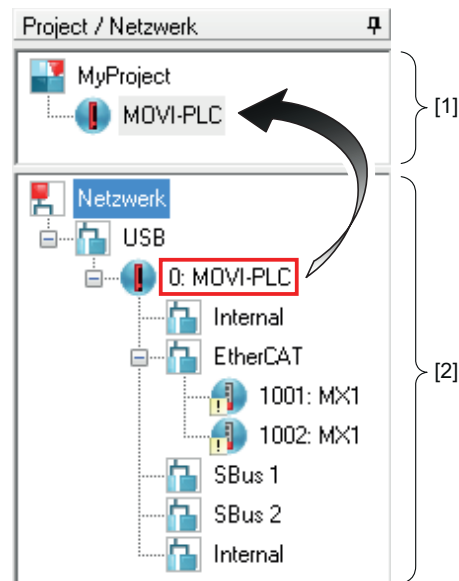


[1]

27021598896943499

- ⇒ The controller is recognized at the engineering interface, and automatically displayed in the network view of MOVITOOLS® MotionStudio.
- ⇒ To highlight that the controller has not been configured yet, an exclamation mark is depicted in front of the controller .

7. To configure the controller, right-click the controller and select [Configure device] in the context menu. Alternatively, mark the controller and move it from the network view [2] to the project view [1] of MOVITOOLS® MotionStudio using drag and drop.

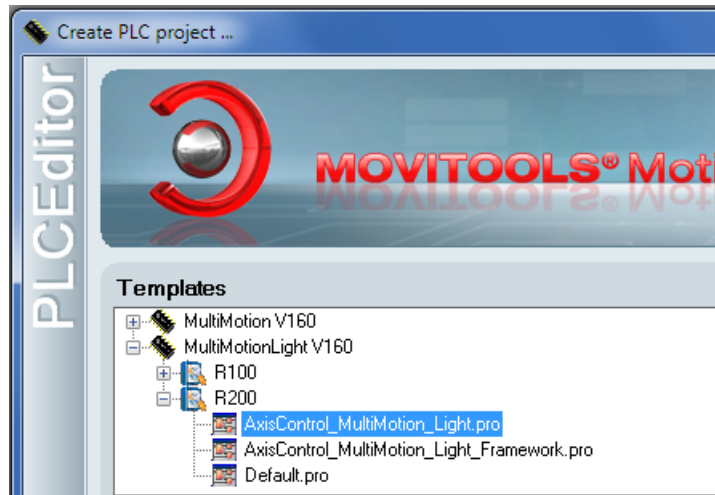


15245260171

8. Enter a device name in the displayed window. The device parameters are uploaded.
- ⇒ The configured device is displayed in the project view.
 - ⇒  This symbol shows that the new unit is configured.
 - ⇒ During project planning, a subfolder with the device name is created in the "Devices" subdirectory of the MOVITOOLS® MotionStudio project folder.
 - ⇒ **INFORMATION:** SEW-EURODRIVE recommends that you do NOT include the firmware in the name, else the save operation will take longer. You can make the relevant setting by choosing [Settings] > [Options] > [Other] from the menu bar.

5.5 Step 2: Creating a PLC Editor project

1. To do so, right-click the controller. From the context menu that opens, select [Programming] > [Create new PLC Editor project].
 - ⇒ The project assistant starts.
2. Select one of the following project templates in the project assistant:
 - ⇒ `AxisControl_MultiMotion.pro` for the program variant *MultiMotion*
 - ⇒ `AxisControl_MultiMotion_Light.pro` for the program variant *MultiMotion Light*



15858989707

INFORMATION



Using project templates

The project template `AxisControl_MultiMotion.pro` for the program variant *MultiMotion* has no additional benefits for the kinematic control.

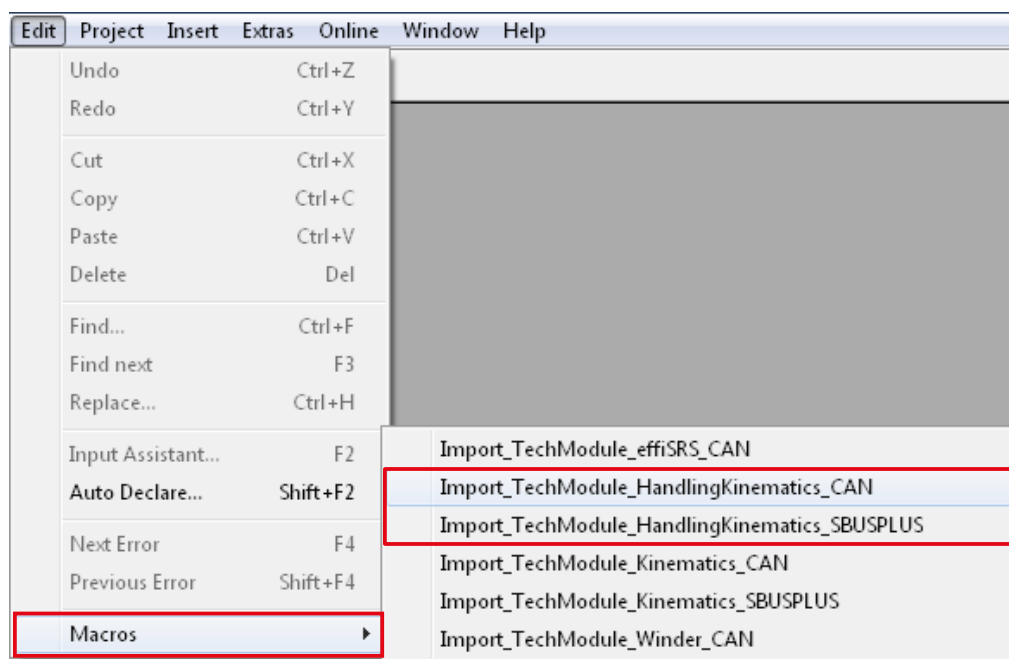
If you want to use a sample application framework, create a project with the project template `xxx_Framework.pro`.

3. Assign a project name. SEW-EURODRIVE recommends to leave the default directory unchanged, as MultiMotion Editor can find the data required for starting the diagnostics more quickly.
 - ⇒ The PLC Editor opens. The selected project template is the basis of the control program to which you can add your own program modules.

5.6 Step 3: Integrating the HandlingKinematics technology module

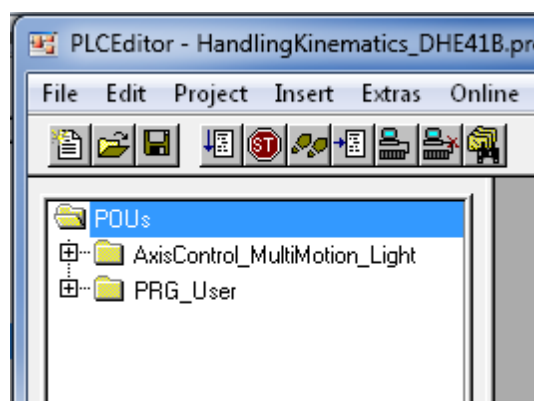
Proceed as follows:

1. Select the menu command [Edit] > [Macros] in the menu bar of the PLC Editor.



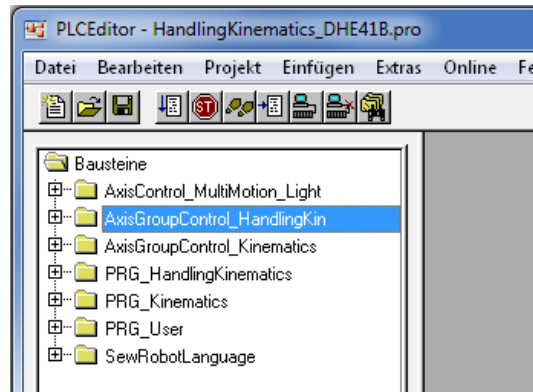
15864469003

2. Select one of the following menu commands, depending on the Bus system connecting the axes to the controller:
 - ⇒ [Import_TechModule_HandlingKinematics_**CAN**]: for "SBus" system bus, based on CAN
 - ⇒ [Import_TechModule_HandlingKinematics_**SBUSPLUS**]: for "SBus^{PLUS}" system bus, based on EtherCAT®
 - ⇒ In case the macro is run successfully, the kinematics folder is added to the PLC Editor project. The following figure shows, which kinematics folder has been added by running the macro.



The directory "Modules" before integrating, or in case of unsuccessful integration of the technology module

16262075275



The directory "Modules" after successful integration of the technology module

16262078987

- ⇒ In case the kinematics folder has not been added while the macro was run, as shown in the figure above, the macro could not be correctly run. One reason for this can be, that the PLC Editor project was saved under a file path that contains spaces, special characters, or umlauts. In this case, the technology module must be manually integrated, as described in the following chapter.
- 3. If you use a customized kinematics model (e.g. Articulated), you must delete the library "04_InterpolatedMotion\MPLCKinematics_UserKin.lib" and add the corresponding UserKin library from SEW-EURODRIVE in the library manager.
- 4. Compile the project in the menu bar via [Project] > [Compile].
- 5. Load the project in the controller of the menu bar via [Online] > [Login].
 - ⇒ In the first step, the project template can be used without any changes.
- 6. Create a boot project in the menu bar via [Online] > [Create boot project].

5.6.1 Manually integrating the HandlingKinematics technology module

Reasons for manual integration

Integrating the technology module via the prefabricated macro is sufficient in most cases. In the following cases, integrating the technology module in the MultiMotion project template via the offered macros is not feasible:

- **Example A: Task configuration is to be maintained.**
When integrating the technology module using macros, the existing task configuration is completely replaced in the PLC Editor project. Instead of creating a new project, it is possible to integrate the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module into an existing PLC Editor project with already changed task configuration. In this case, you can maintain the task configuration and only integrate the libraries and open programs of the "HandlingKinematics" application module. The task configuration must be manually extended in this case.
- **Example B: Macro could not be successfully run. The kinematics folder could not be added to the PLC Editor project.**

Manually integrating the technology module

In both cases, proceed as follows:

1. Select the menu command [Project] > [Import] in the menu bar of the PLC Editor.

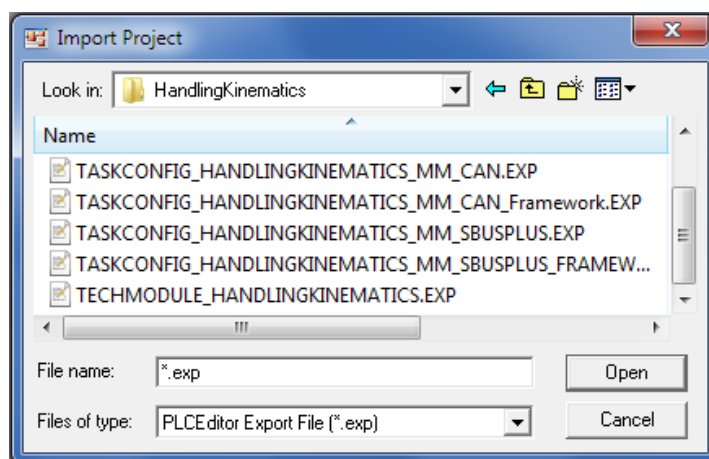
2. Browse to the directory ...\"Your PLC-Project folder\"...\\TechModules\\HandlingKinematics, that you have created in step 2 (→ 22).

INFORMATION



For the import, make sure to navigate to the PLC Editor project directory that you created in step 2: Creating a PLC Editor project (→ 22). The window that appears does not navigate to this directory automatically. If you import the wrong files, you will either not be able to compile the PLC Editor project without errors or it will not work correctly.

⇒ The found directory is as follows.



15865277835

3. Proceed as follows, depending on the respective case:

Example A	Example B
To only add libraries and open source code from HandlingKinematics, while the previous task configuration is maintained:	To integrate the entire technology module, including task configuration:
- Only select the file <code>TECHMODULE_HANDLINGKINEMATICS.EXP</code>. - Manually extend the task configuration. For detailed information, refer to chapter "Task configuration" (→ 32). → The manual integrating is complete.	- In addition to the file <code>TECHMODULE_HANDLINGKINEMATICS.EXP</code>, press and hold the control key while you select the task configuration suitable to your project, for example <code>TaskConfig_HandlingKinematics_xxx.EXP</code>: <code>xxx_CAN.EXP</code>: for use of the system bus "SBus", based on CAN <code>xxx_SBUSPLUS.EXP</code>: for use of system bus "SBus^{PLUS}", based on EtherCAT® <code>xxx_Framework.EXP</code>: for use of the application framework - Confirm your selection by clicking [Open]. The relevant elements are integrated. - The task configuration is overwritten. Confirm the displayed message with [Yes] or [Yes, all]. → The manual integrating is complete.

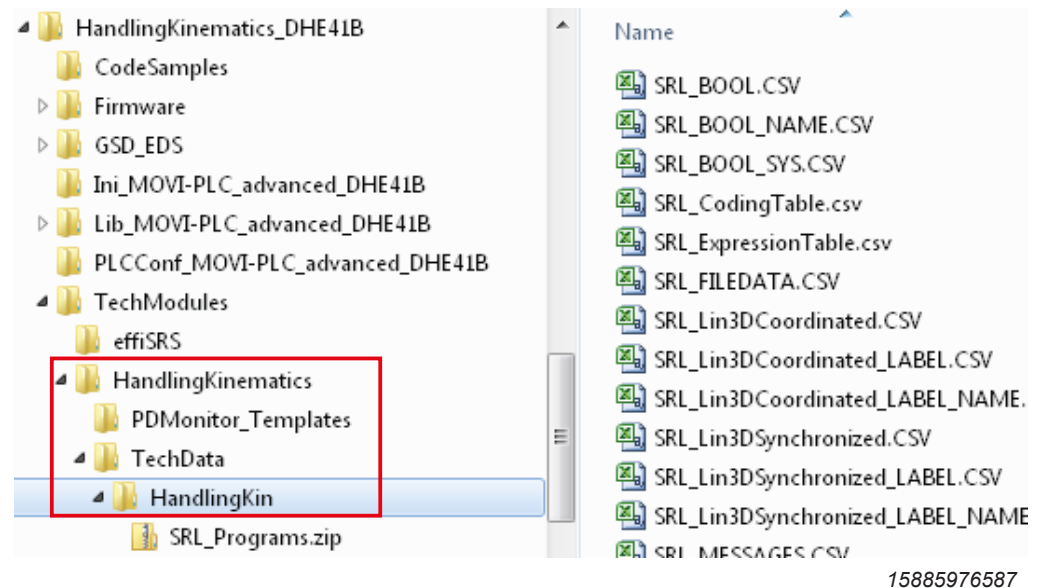
4. Compile the project in the menu bar via [Project] > [Compile].
5. Load the project in the controller of the menu bar via [Online] > [Login].

5.7 Step 4: Copying SRL files to the controller memory card

To use the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module on the MOVI-PLC® controller, it is necessary to copy the SRL files (SRL = SEW Robot Language) to the controller memory card. The SRL files are contained in the directory of the PLC Editor project.

Proceed as follows:

1. On the engineering PC hard disk, open the folder where you have created the PLC Editor project (→ 22). In case you maintained the default directory for creating the PLC Editor project, the PLC Editor project can be found in the MotionStudio project directory via `MyMotionStudioProject/Devices/MyPLC/PLCEditor/MyHandlingKinematicsProject`.



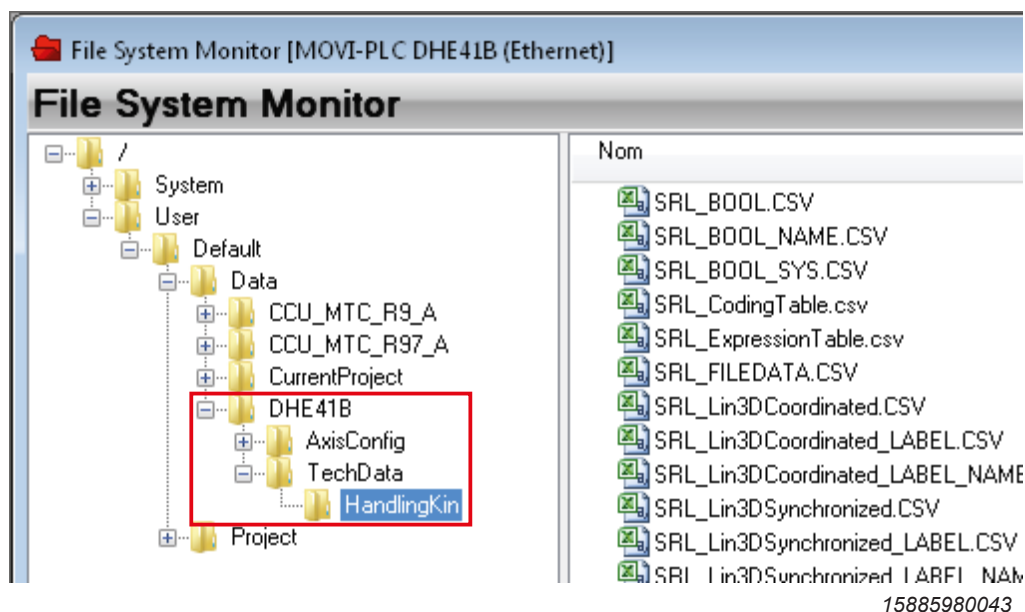
15885976587

INFORMATION



To copy the folder "TechData", navigate to the directory of the PLC project that you created in step 2 (→ 22). The technology module will not function properly, if the wrong files are copied.

2. Copy the entire folder "TechData" from `TechModules\HandlingKinematics`. Use either File System Monitor or a card reader to copy the folder to the following folder on the controller memory card: `User/Default/Data/"Controller-Typ"`.
 - ⇒ To copy the folder "TechData" **via the card reader**, read the controller memory card using a card reader.
 - ⇒ To copy the folder "TechData" **via File System Monitor**, right-click on the controller in MOVITOOLS® MotionStudio, and select the menu command [Diagnostics] > [File System Monitor]. Copying via File System Monitor Folder "TechData"
 - ⇒ The following figure shows the required storage location for the folder "TechData" on the memory card.



5.8 Step 5: Starting up the single axes

INFORMATION

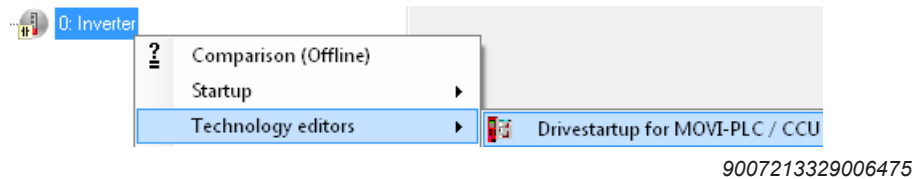


This step is not required, if you only program offline using simulated axes.

- In this case, continue the startup with step 6: Configuring the kinematic model (→ 29).

Proceed as follows:

1. Select the inverter in the network view of MOVITOOLS® MotionStudio.



2. Right-click to open the context menu of the inverter.
3. Select the menu command [Technology editor] > [Drive Startup for MOVI-PLC®/CCU].
⇒ Drive Startup for MOVI-PLC®/CCU is started.
4. Follow the instructions of the wizard.

INFORMATION



A common configuration error is incorrect setting of baud rate and SBus addresses.

- The baud rate must be set identically for all connected devices.
- The SBus address must always differ for the various devices.

5.9 Step 6: Configuring the kinematic model

5.9.1 Adding the HandlingKinematics technology module

Proceed as follows:

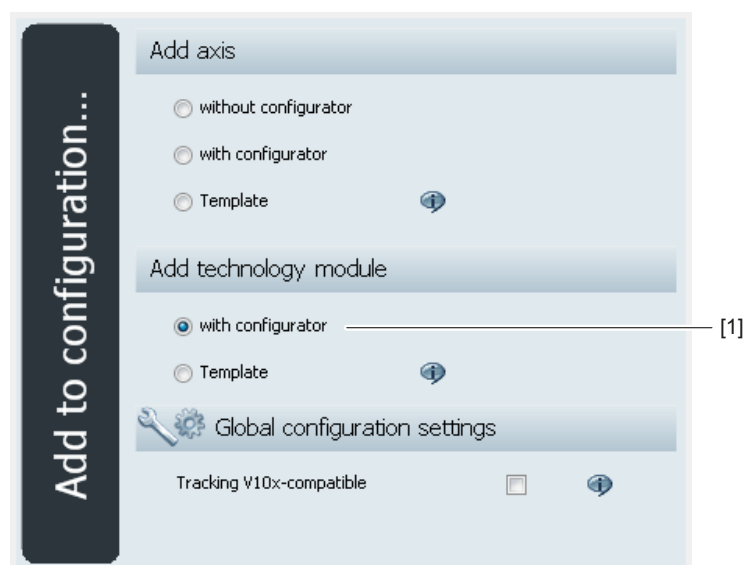
1. On the initial screen of the MultiMotion Editor, click the button [1].



[1]

15895379211

2. In the displayed window under "Adding technology module", select the setting "with Configurator" [1] and click [Next].



[1]

15895474443

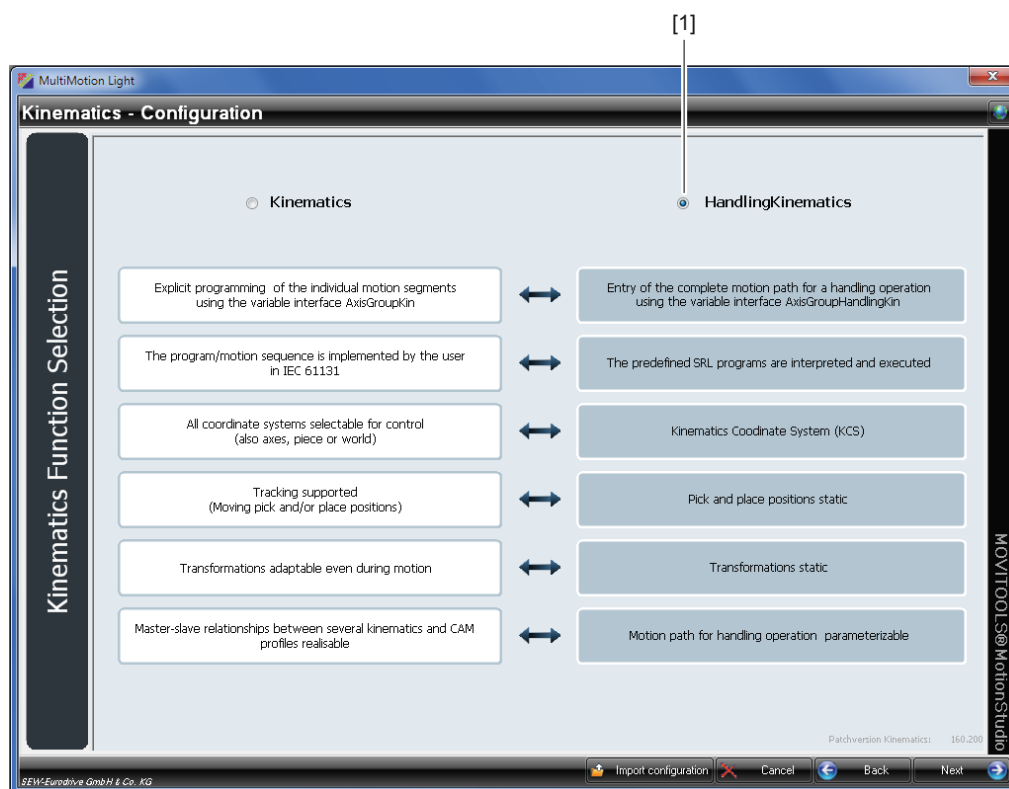
3. Select the "Kinematics" technology module [1] and click [Next].



15895478155

⇒ The Kinematics Configurator opens.

4. Select the "HandlingKinematics" technology module [1] and click [Next].



15895482123

5. Configure the "HandlingKinematics" technology module as described below.

5.9.2 Configuring the "HandlingKinematics" technology module



INFORMATION

The further configuration of the "HandlingKinematics" technology module corresponds to the configuration of the "HandlingKinematics" CCU application module.

For detailed information on the following topics, refer to the manual "HandlingKinematics CCU application module":

- Kinematics selection
- Functionality
- Configuring the axes:
 - General settings
 - Encoder
 - Scaling
 - Homing parameters
 - Touchprobe function
 - System limits of the axes
 - Axis-by-axis jog
 - Digital inputs
- Configuration of the kinematic model:
 - Kinematic parameters
 - Kinematic limitations
 - Cartesian software limit switch
- Configuration of the motion parameters:
 - Cartesian motion parameters
 - Continuous path motion parameters
 - Segment parameter sets

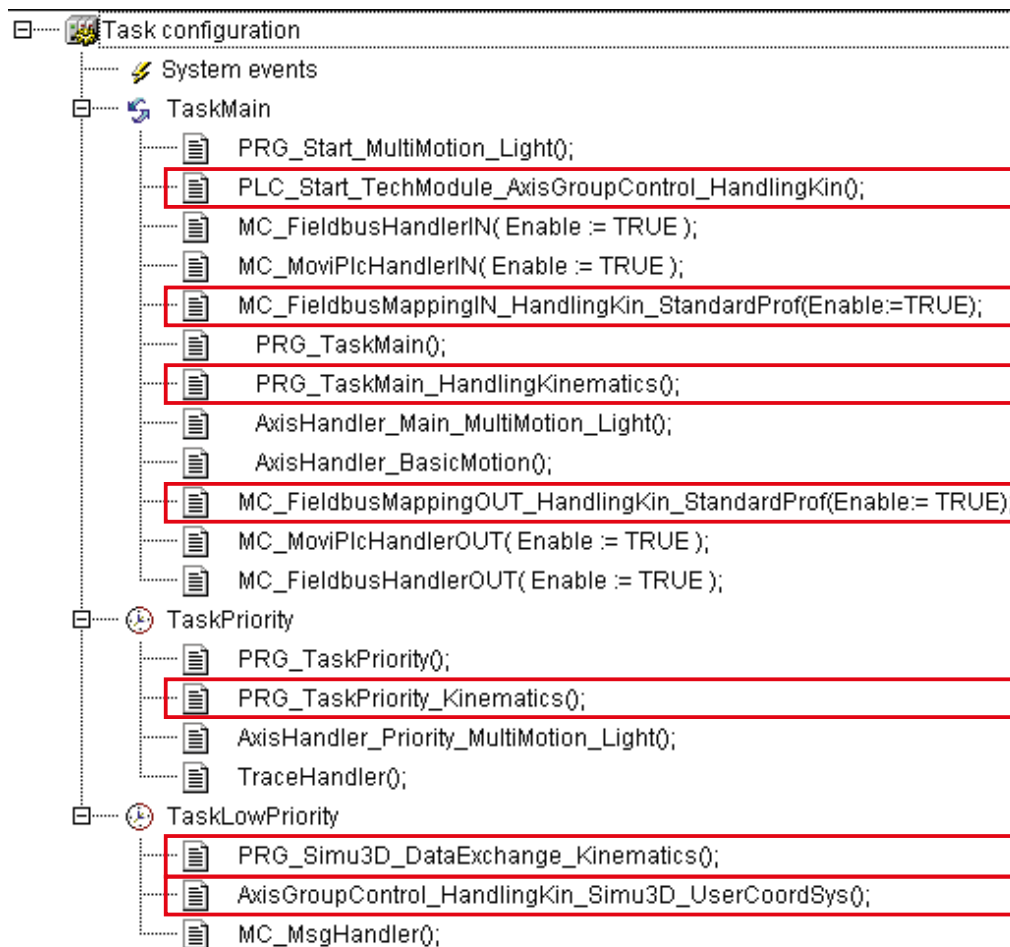
When you have successfully completed the configuration, the Kinematics Configurator closes and adds the configured technology module. You can then open and edit it any time using the MultiMotion Editor.

6 MOVI-PLC® program

6.1 Task configuration

A suitable task configuration is automatically added when a new PLC Editor project is created, see chapter "Step 2: Creating a PLC Editor project" (→ 22) and "Step 3: Integrating the HandlingKinematics technology module" (→ 23).

The following task entries are required for the technology module. When the PLC Editor project is not newly created, but only needs to be expanded, you can also add these task entries manually.

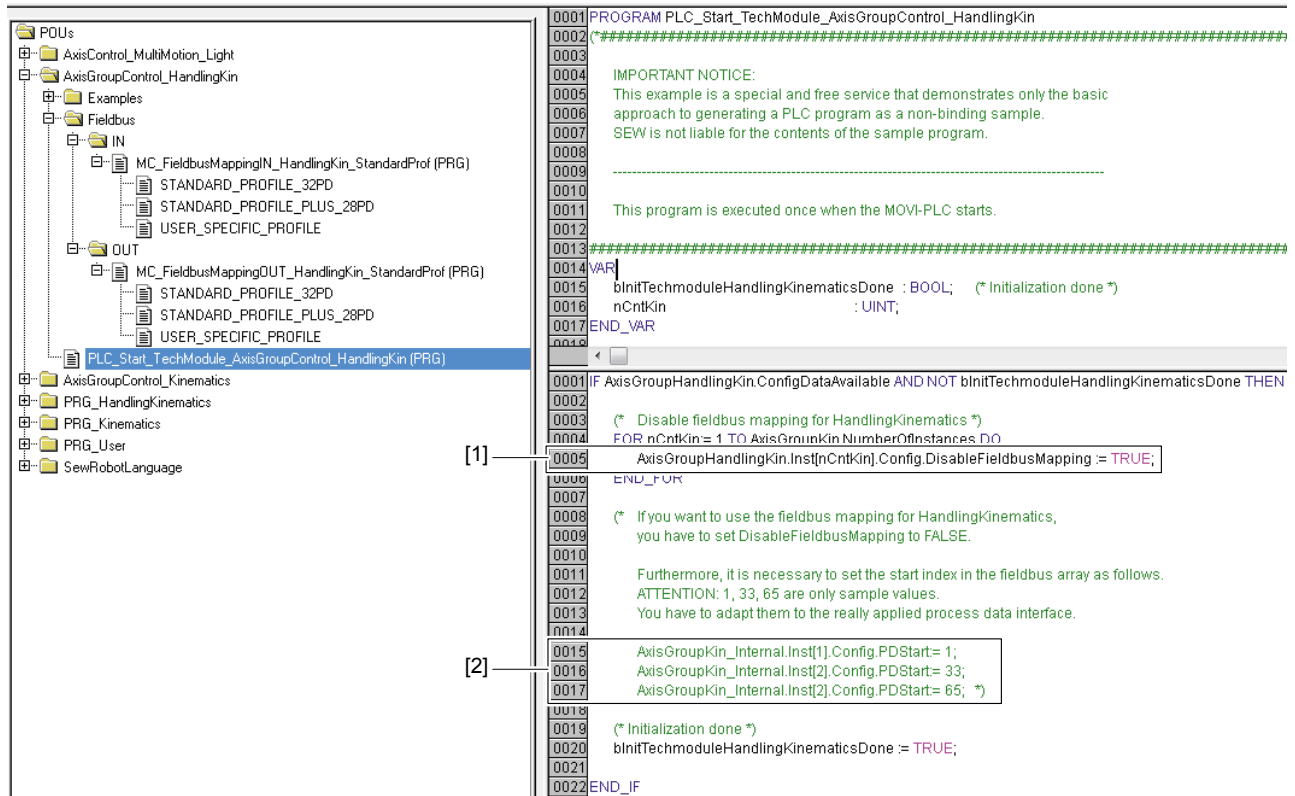


15918673291

6.2 Control via fieldbus

6.2.1 Activating the fieldbus mapping

For standard use of the fieldbus mapping, as in the "HandlingKinematics" CCU application module, activating the fieldbus mapping in the program PLC_Start_TechModule_AxisGroupControl_HandlingKinematics will suffice.



```

0001 PROGRAM PLC_Start_TechModule_AxisGroupControl_HandlingKin
0002 (*#####*)
0003
0004 IMPORTANT NOTICE:
0005 This example is a special and free service that demonstrates only the basic
0006 approach to generating a PLC program as a non-binding sample.
0007 SEW is not liable for the contents of the sample program.
0008
0009 -----
0010
0011 This program is executed once when the MOVI-PLC starts.
0012
0013 #####
0014 VAR
0015   bInitTechmoduleHandlingKinematicsDone : BOOL; (* Initialization done *)
0016   nCntKin : UINT;
0017 END_VAR
0018
0019 IF AxisGroupHandlingKin.ConfigDataAvailable AND NOT bInitTechmoduleHandlingKinematicsDone THEN
0020
0021   (* Disable fieldbus mapping for HandlingKinematics *)
0022   FOR nCntKin = 1 TO AxisGroupKin.NumberOfInstances DO
0023     AxisGroupHandlingKin.Inst[nCntKin].Config.DisableFieldbusMapping := TRUE;
0024   END_FOR
0025
0026   (* If you want to use the fieldbus mapping for HandlingKinematics,
0027    you have to set DisableFieldbusMapping to FALSE.
0028
0029   Furthermore, it is necessary to set the start index in the fieldbus array as follows.
0030   ATTENTION: 1, 33, 65 are only sample values.
0031   You have to adapt them to the really applied process data interface.
0032
0033   AxisGroupKin_Internal.Inst[1].Config.PDStart:= 1;
0034   AxisGroupKin_Internal.Inst[2].Config.PDStart:= 33;
0035   AxisGroupKin_Internal.Inst[2].Config.PDStart:= 65; *)
0036
0037   (* Initialization done *)
0038   bInitTechmoduleHandlingKinematicsDone := TRUE;
0039 END_IF
0040
  
```

15918677003

No.	Description
[1]	In line 5, you can activate or deactivate the fieldbus data mapping on the variable interface <i>AxisGroupHandlingKin</i>: - Activate fieldbus mapping: – <i>AxisGroupHandlingKin.Inst[nCntKin].Config.DisableFieldbusMapping :=FALSE</i> - Deactivate fieldbus mapping: – <i>AxisGroupHandlingKin.Inst[nCntKin].Config.DisableFieldbusMapping :=TRUE</i>
[2]	In line 15 – 17, you can set, at what point in the fieldbus interface the data for the relevant kinematic instance will begin. This start index is set for each used kinematic instance (for example, copy line 15 from the comment field).

6.2.2 Using the process data monitor

The most important part of the process data monitor is to display fieldbus input data and fieldbus output data that are exchanged between the controller and the higher-level controller.

The process data monitor offers the following possibilities:

- **Structure and naming of input and output data**

The transmitted input and output data can be structured and named.

- **Control mode for comprehensive testing of the fieldbus interface**

The fieldbus interface can be tested in control mode, even if no higher-level controller is available.

- **Using a template for the fieldbus interface**

The structure of the fieldbus interface can be depicted user-specific in the process data monitor. This can be saved for later use as a template.

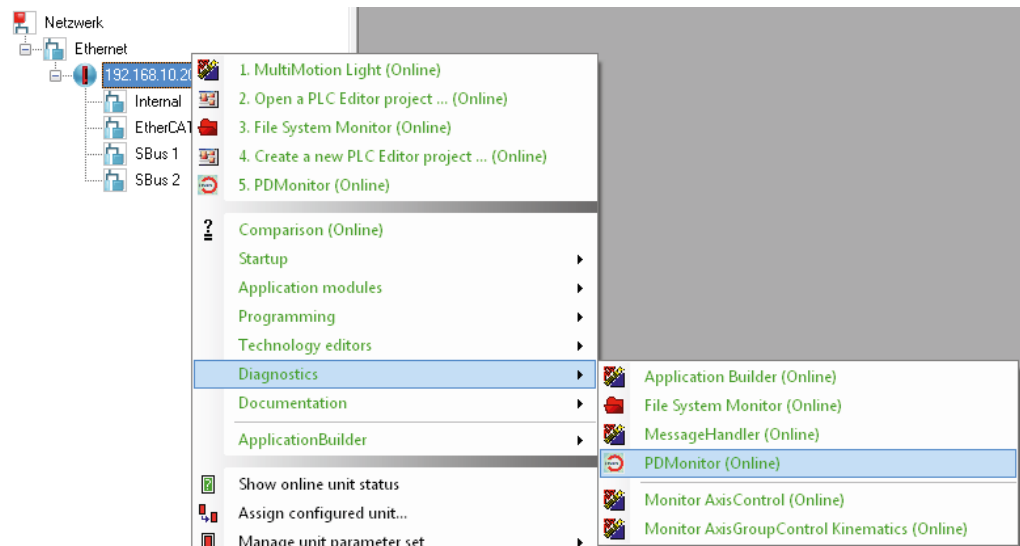
- **Using the 4 default templates**

The process data monitor contains 4 default templates that you can download directly. The templates correspond to the 4 process data profiles of the "HandlingKinematics" application module.

Starting the process data monitor

Proceed as follows:

1. Select the controller in the network view of MOVITOOLS® MotionStudio.
2. Right-click to open the context menu of the controller.



15918706315

3. Select the menu command [Diagnostics] > [PD monitor].

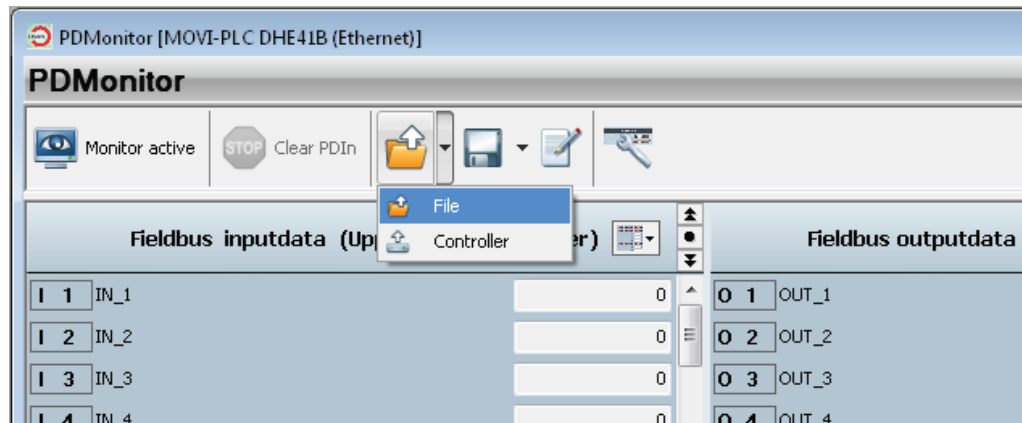
⇒ The process data monitor is started.

Selecting the process data monitor template

Proceed as follows:

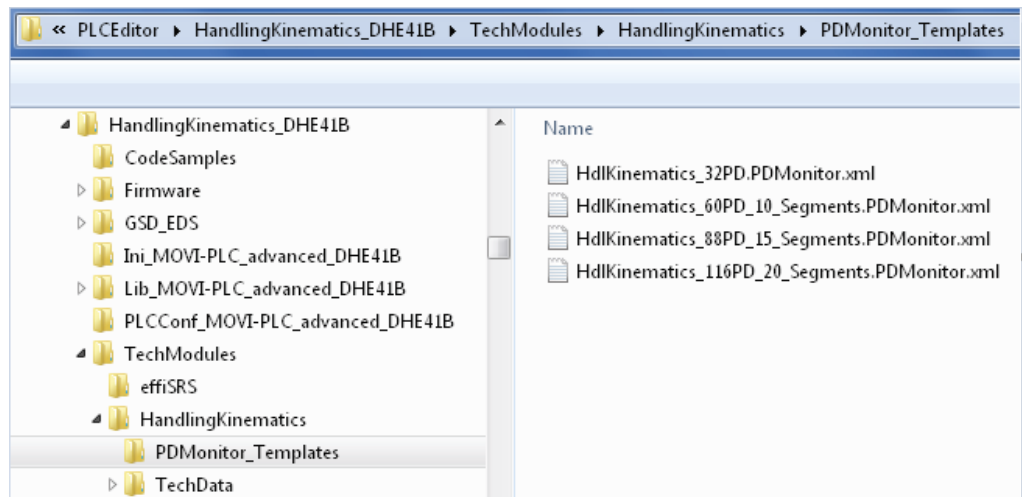
- ✓ Process data monitor is open.

1. In the menu bar, click the button [File].



15918710027

- ⇒ The templates are contained in the directory of the PLC Editor project.
2. Navigate to the following folder in your PLC Editor project:
 - ⇒ MyMotionStudioProject/Devices/MyPLC/PLCEditor/MyHandlingKinematicsProject/TechModules/HandlingKinematics/PDMonitorTemplates



15918713739

- ⇒ The 4 default templates correspond to the 4 process data profiles of the "HandlingKinematics" application module.
- ⇒ If as standard you use the same fieldbus mapping that is used in the "HandlingKinematics" CCU application module, you can use the templates without adjustment.

INFORMATION



You can also adjust the fieldbus mapping to your requirements or leave it out completely.

If you do not use fieldbus mapping, operate the "HandlingKinematics for MultiMotion/ MultiMotion Light" technology module directly in the program. Use the following 2 global variable interfaces, described in detail in chapter "Variable interfaces" (→ 42):

- *AxisGroupHandlingKin*
- *AxisGroupHandlingKinProfile*

For detailed information on adjusting the fieldbus mapping, and on templates, refer to chapter "Changing the existing fieldbus mapping or integrating a user fieldbus mapping" (→ 60).

6.3 Example for a sequential program in the controller

INFORMATION

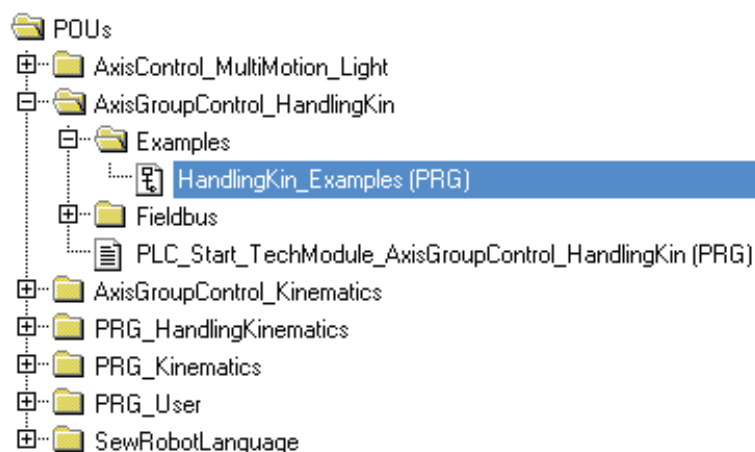


The kinematic control is performed exclusively via the global variable interfaces *AxisGroupHandlingKin* and *AxisGroupHandlingKinProfile*.

The following variables and variable interfaces must **NOT** be written:

- The variable interface *AxisGroupKin.Inst[.]* of the lower-level "Kinematics" technology module with the same instance number
- The variable *AxisInterface* of the MultiMotion axes referring to this kinematic instance

The PLC Editor project created in step 2 (→ 22) during startup contains an example of the sequential program for a handling task.



15918743051

The example illustrates how to program the variable interface *AxisGroupHandlingKin*.

Here, 2 pick and place programs are performed alternately. These programs can only be run by kinematic models with the degree of freedom X, Y, Z and A. To run these programs, for example the kinematic model *CARTESIAN_GANTRY_LLLR_XYZA_M10* with default settings and axis simulation can be configured.

22323414/EN – 12/2015

In the example, the path segment target poses are hard-coded in the step "Init". The program 4 (CP with synchronized change in orientation) is used. This is selected in the step "Select_Program_Mode". The example shows, how to handle an interruption in the program sequence. In case of an interruption (Output bit *ApplicationReady* = FALSE), change to step "Disruption". Based on this step is decided, if BackToPath is required to continue the program (e.g. in case the inverter enable was withdrawn and thus no stop was performed on the path) or the program sequence can be continued directly (e.g. if no path deviation is detected after restart).

The code positions required for the Touchprobe functions *Touchprobe Measure* or *Touchprobe Motion* can be activated via the variables.

INFORMATION

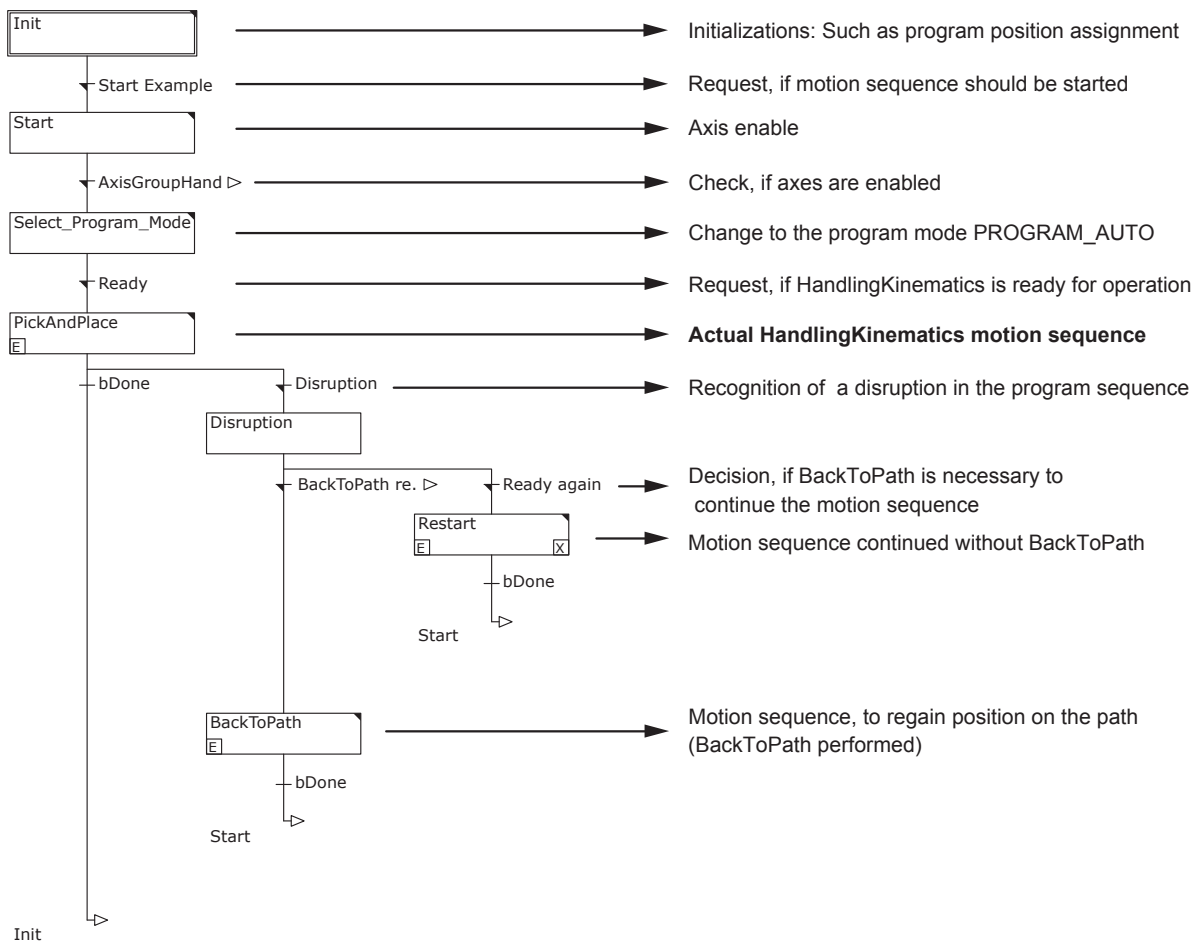


This example is a special and free service that demonstrates only the basic procedure of generating a sequential program as a non-binding example. SEW-EURODRIVE is not liable for the contents of the sample program.

To ensure a correct application and conditions, the following must be explicitly designed, implemented and tested:

- Positions
- Procedure
- Handling faults
- Home drives
- Behavior in special situations
- Managing operating modes
- and others

Example for sequential program



15918750475

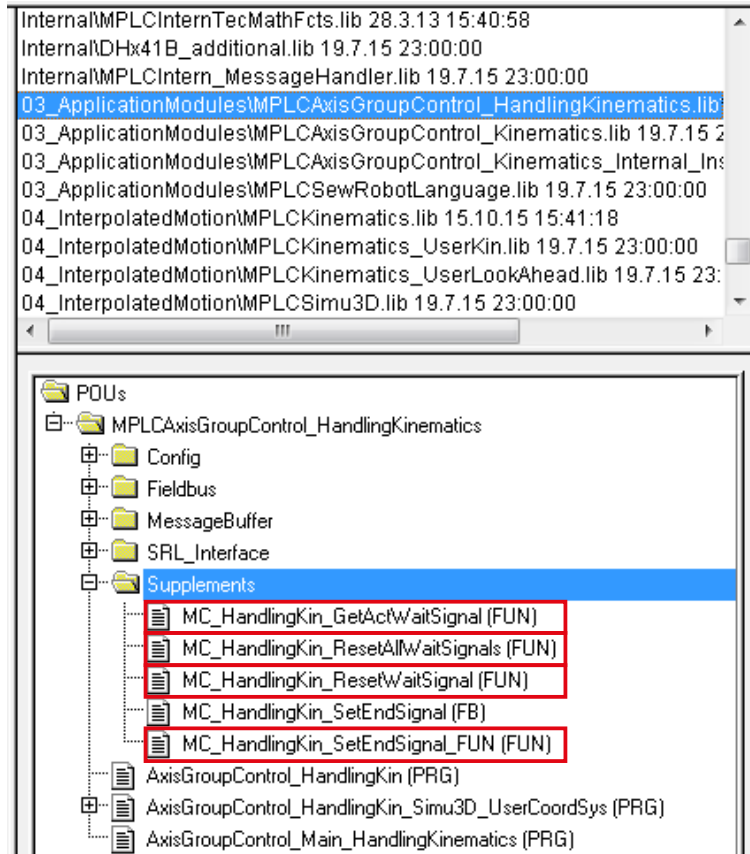
The following variables for control of the sequential program must be declared in the module header.

Variable	Description
<i>bEnable</i>	Enabling the program processing
<i>iOverride</i>	Setting the override (speed of 0 – 100%)
<i>bContinueAfterDisruption</i>	Continuing the path movement after an interruption, for example after emergency switching off (use BackToPath function)
<i>bUseTouchProbe_Motion</i>	Activating the Touchprobe function <i>Touchprobe Motion</i> in the sequential program
<i>bUseTouchProbe_Measure</i>	Activating the Touchprobe function <i>Touchprobe Measure</i> in the sequential program

For the exact processing of the control variables, refer to the source code of the sequential program.

6.4 Functions for application programming

The following functions are available to simplify handling the single wait and end signals. These functions can be found in the folder "Supplements" in the library *03_ApplicationModules\MPLCAxisGroupControl_HandlingKinematics.lib*.



16143380747

Function	Description
MC_HandlingKin_GetActWaitSignal	This function determines, if the kinematic model has stopped and at what wait point it is currently located.
MC_HandlingKin_ResetAllWaitSignals	This function deactivates all wait points.
MC_HandlingKin_ResetWaitSignal	This function deactivates a certain wait signal.
MC_HandlingKin_SetEndSignal_FUN	This function activates a certain end signal.

The wait and end signals can still be directly controlled and read via the variable interface.

6.4.1 MC_HandlingKin_GetActWaitSignal

This function determines, if the kinematic model has stopped and at what wait point it is currently located.

The input variable *iKinInstanceNumber* must transfer which kinematic instance you want to observe. If a kinematic instance of HandlingKinematics is used on the MOVIPLC® controller, transfer the value "1" to the input variable *iKinInstanceNumber*. In case of more than one kinematic instances on one controller, the value "1" must be transferred for the first kinematic instance, "2" must be transferred for the second kinematic instance etc.

If the return value of the function is "0", the kinematic model is not currently in a wait signal. This for example applies, when the kinematic model is still in motion or the program is already complete.

A return value larger than "0" describes, at which path point the kinematic model currently waits. The function *ResetWaitSignal* allows for forwarding the affected wait signal.

6.4.2 MC_HandlingKin_ResetAllWaitSignals

This function deactivates all wait points. The variables *AxisGroupHandlingKinProfile.Inst[iKinInstanceNumber].Standard.In.PrgControlAdditional.Wait1-20* are set to FALSE. You can use this for example to ensure that no wait signal is still active before a new program is started. When this function is opened, the desired wait signals can be activated if necessary, to ensure that no wait signals remain active from the previous program sequences.

The input variable *iKinInstanceNumber* must transfer which kinematic instance you want to observe. If a kinematic instance of HandlingKinematics is used on the MOVI-PLC® controller, transfer the value "1" to the input variable *iKinInstanceNumber*. In case of more than one kinematic instances on one controller, the value "1" must be transferred for the first kinematic instance, "2" must be transferred for the second kinematic instance etc.

6.4.3 MC_HandlingKin_ResetWaitSignal

This function deactivates a certain wait signal. The variable *AxisGroupHandlingKinProfile.Inst[iKinInstanceNumber].Standard.In.PrgControlAdditional.Wait"X"* is set to FALSE. The input variable *iWaitIndex* must be used to transfer which wait signal is deactivated. All other wait signals are not changed.

The input variable *iKinInstanceNumber* must transfer which kinematic instance you want to observe. If a kinematic instance of HandlingKinematics is used on the MOVI-PLC® controller, transfer the value "1" to the input variable *iKinInstanceNumber*. In case of more than one kinematic instances on one controller, the value "1" must be transferred for the first kinematic instance, "2" must be transferred for the second kinematic instance etc.

The function responds with TRUE when the deactivation was successful. If the deactivation was not successful, for example when the value of the input variable *iWaitIndex* is not between 1 – 20, the function responds with FALSE.

6.4.4 MC_HandlingKin_SetEndSignal_FUN

This function activates a certain end signal. The variable *AxisGroupHandlingKinProfile.Inst[iKinInstanceNumber].Standard.In.PrgControlAdditional.End"X"* is set to TRUE. The input variable *NumberOfSegments* must be used to transfer which end signal is activated. All other end signals are set to FALSE.

The HandlingKinematics SRL program is finished after the number of path segments determined via the variable *iNumberOfSegments*.

For the end signal to become valid, this function must be opened before the HandlingKinematics SRL program is started. If the activation was successful, the function responds with TRUE. If the activation was not successful, for example when the value of the input variable *iNumberOfSegments* is not between 1 – 20, the function responds with FALSE.

The input variable *iKinInstanceNumber* must transfer which kinematic instance you want to observe. If a kinematic instance of HandlingKinematics is used on the MOVI-PLC® controller, transfer the value "1" to the input variable *iKinInstanceNumber*. In case of more than one kinematic instances on one controller, the value "1" must be transferred for the first kinematic instance, "2" must be transferred for the second kinematic instance etc.

7 Variable interfaces

INFORMATION



The abbreviation RTRIG (Rising Trigger) indicates that a rising edge (FALSE→TRUE) of the bit is present.

7.1 Variable interface AxisGroupHandlingKin.Inst[..]

The global variable structure *AxisGroupHandlingKin* represents the basic part of the variable interface that cannot be changed. This contains general control and status variables of HandlingKinematics.

Variable: *ConfigDataAvailable*
Data type: BOOL

The configuration data and default values for input variables for all kinematic instances and all axes were read from the memory card and the corresponding variables were assigned.

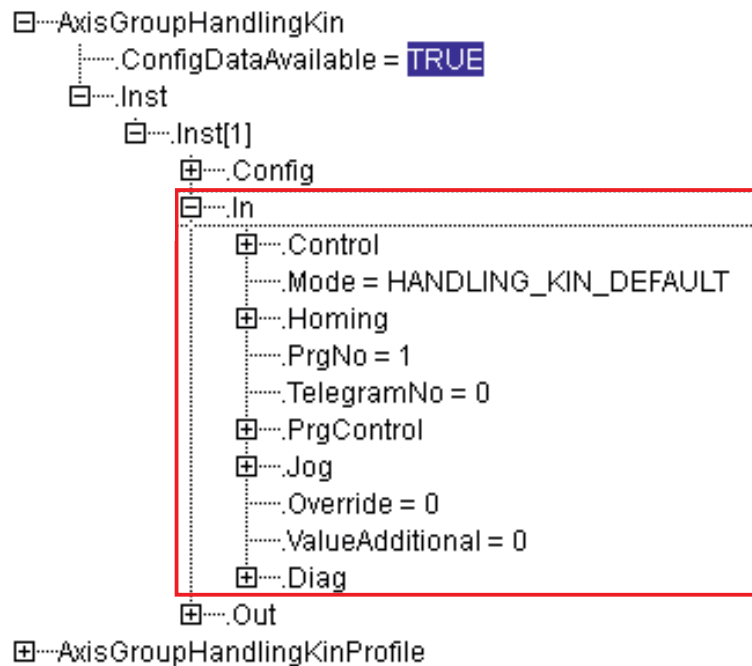
INFORMATION: As long as *ConfigDataAvailable* = FALSE, the application program must NOT be executed.

Variable: *Inst*
Data type: ARRAY [1..] OF ST_AxisGroupHandlingKin_Instance

The variable structure contains configuration, input and output data of the kinematic instances. The following variables are available for each kinematic instance.

<i>Config</i>	ST_AxisGroupHandlingKin_Configuration
<i>In</i>	ST_AxisGroupHandlingKin_In
<i>Out</i>	ST_AxisGroupHandlingKin_Out

7.1.1 Control variables: AxisGroupHandlingKin.Inst[..].In



15923988235

Data type: ST_AxisGroupHandlingKin_In**Library:** 03_ApplicationModules\MPLCAxisGroupControl_HandlingKinematics.lib

Variable: Control Data type: ST_AxisGroupHandlingKin_IN_Control			Correspondent in the PD interface
The variable structure contains the following control variables that are independent of the operating mode.			
<i>Inhibit</i>	BOOL	TRUE: Activates the controller inhibit on all axes.	I1':0 <i>Controller Inhibit</i>
<i>Enable_RapidStop</i>	BOOL	TRUE: Enables the interpolated position control of the kinematic axes. FALSE: The kinematic model performs a rapid stop. Initially, the model is kinematically decelerated with the configured rapid stop ramps. Once the standstill is reached, the inverter enable is automatically removed.	I1':1 <i>Enable/Rapid Stop</i>
<i>FeedEnable</i>	BOOL	TRUE: Motion enable (feed) of the kinematic model. FALSE: The kinematic model stops on the path.	I1':2 <i>Feed Enable</i>
<i>Simu3DEnable</i>	BOOL	TRUE: 3D simulation activated. FALSE: 3D simulation not run.	—
<i>Simu3DPenOn</i>	BOOL	TRUE: Pen in 3D simulation is activated	I1':4 <i>Simulation 3D Pen</i>
<i>Simu3DDeleteAll-PenLines</i>	BOOL	RTRIG: Lines in 3D simulation are deleted	I1':5 <i>Simulation 3D Delete Lines</i>

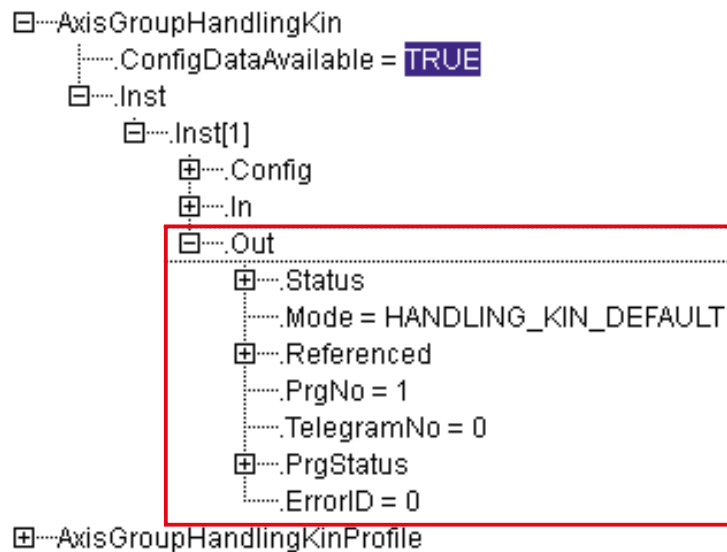
Variable: <i>Control</i> Data type: <i>ST_AxisGroupHandlingKin_IN_Control</i>			Correspondent in the PD interface
<i>Simu3DDisplayProgramPositions</i>	BOOL	TRUE: The path points defined in the program are displayed in coordinate systems in the 3D simulation. FALSE: The path points are not displayed in the 3D simulation.	–
<i>ErrorReset</i>	BOOL	RTRIG: Resets errors. In addition, the output of the maximum reached motor speed is reset to 0.	I1':6 <i>Error Reset</i>
<i>DisplayOfActMotorSpeed</i>	BOOL	The variables control the fieldbus mapping of the output variables on the fieldbus interface. INFORMATION: Only relevant, if HandlingKinematics is operated at an PLC via a fieldbus. TRUE: The signal O16' – O19' <i>Motor Speed</i> displays the current motor speed. FALSE: The signal O16' – O19' <i>Motor Speed</i> displays the highest motor speed reached last.	I1':7 <i>Display Current Motor Speed</i>
<i>DisplayOfTouchprobeMeasure</i>	BOOL	The variables control the fieldbus mapping of the output variables on the fieldbus interface. INFORMATION: Only relevant, if HandlingKinematics is operated at an PLC via a fieldbus. TRUE: The signal O7' – O10' <i>Position/Touchprobe Measure</i> displays the measured pose (XYZA) referring to the kinematics coordinate system (KCS) ¹⁾ that was last determined in the first path segment of <i>Touchprobe Measure</i> . FALSE: The signal O7' – O10' <i>Position/Touchprobe Measure</i> displays the current pose (X,Y,Z,A) referring to the kinematics coordinate system (KCS) ¹⁾ .	I1':8 <i>Display Touchprobe Measure</i>
<i>IgnoreMotorSpeedLimitSwitch</i>	BOOL	TRUE: Motor speed limitation is ignored. The variable is still evaluated and displayed in the signal <i>AxisGroupHandlingKin.Inst[.].Out.Status.MaximalMotorSpeedCrossed</i> .	I1':14 <i>Ignore Over-speed</i>
<i>DisableSWLSWork-space</i>	BOOL	TRUE: Kinematic, Cartesian and axis limits are ignored, that means if these limits are exceeded no error is issued and no rapid stop is performed. The state of the software limit switches is still evaluated and displayed in the signal <i>AxisGroupHandlingKin.Inst[.].Out.Status.Work-spaceLeft</i> .	I1':15 <i>Ignore Work Envelope Monitoring</i>

1) KCS = Kinematics coordinate system

Variable: <i>Mode</i> Data type: MC_AxisGroupHandlingKin_Mode			Correspondent in the PD interface
The variables for selecting the operating mode. The following input values are available:			I2':0 – 4 <i>Operating Mode</i> :
- HANDLING_KIN_DEFAULT (Operating mode inactive <i>Default</i>) - HANDLING_KIN_JOG_AXIS (Axis-by-axis jog mode <i>Jog Axis</i>) - HANDLING_KIN_JOG_CART (Cartesian jog mode <i>Jog Cartesian</i>) - HANDLING_KIN_HOMING (Referencing mode <i>Homing</i>) - HANDLING_KIN_PROGRAM_AUTO (Program mode <i>Program Auto</i>) - HANDLING_KIN_PROGRAM_STEP (Program mode <i>Program Step</i>)			<i>Jog Axis</i> <i>Jog Cartesian</i> <i>Homing</i> <i>Program Auto</i> <i>Program Step</i>
Variable: <i>Homing</i> Data type: ST_AxisGroupHandlingKin_IN_Homing			Correspondent in the PD interface
The variable structure contains the following control variables for referencing axes.			
<i>Start</i>	ARRAY [1..8] OF BOOL	TRUE: Starts referencing the respective axis in referencing mode.	I2':8 – 15 <i>Start Referencing</i>
Variable: <i>PrgNo</i> Data type: UINT			Correspondent in the PD interface
The number of the selected programs in program mode <i>Program Auto/Step</i> :			I3':High byte <i>Program Number</i>
1: TARGET AXIS 2: TARGET CART 3: LINEARLY coordinated 4: LINEARLY synchronized			
INFORMATION: If a program number other than 1 – 4 is assigned, an error is issued. In case of the application-specific user program, no error is issued.			
Variable: <i>TelegramNo</i> Data type: UINT			Correspondent in the PD interface
The telegram number ensures consistency in data transmission on the fieldbus.			I3':Low-Byte <i>Telegram Number</i>
INFORMATION: Only relevant, if HandlingKinematics is operated at a PLC via a fieldbus. In this case, the telegram number must be used to ensure the telegram consistency on the fieldbus: Only optional for process data profile 1, required for process data profiles 2 – 4.			
For details on consistent data transmission, refer to the manual "HandlingKinematics CCU application module".			
Variable: <i>PrgControl</i> Data type: ST_AxisGroupHandlingKin_IN_ProgramCtrl			Correspondent in the PD interface
The variable structure contains the following variables for controlling the program processing in program mode <i>Program Auto/Step</i> . For detailed information refer to the manual "HandlingKinematics CCU application module".			
<i>Init</i>	BOOL	RTRIG: (New) Initialization of the program	I4':0 <i>Program Init</i>
<i>Start</i>	BOOL	RTRIG: The program is executed	I4':1 <i>Program Start</i>
<i>Stop</i>	BOOL	TRUE: Stops the program	I1':2 <i>Program Stop</i>

Variable: <i>Jog</i> Data type: ST_AxisGroupHandlingKin_IN_Jog			Correspondent in the PD interface
The variable structure contains the following control variables for jogging the axes in jog mode <i>Jog Axis</i> or for jogging the Cartesian degrees of freedom in jog mode <i>Jog Cartesian</i> .			
<i>Positive</i>	ARRAY [1..8] OF BOOL	TRUE: - In jog mode <i>Jog Axis</i> (HANDLING_KIN_JOG_AXIS): Jog in positive direction A1, A2, A3 or A4 - In jog mode <i>Jog Cartesian</i> (HANDLING_KIN_JOG_CART): Jog in positive direction X, Y, Z or A	I5':0 – 7 <i>Jog positive</i>
<i>Negative</i>	ARRAY [1..8] OF BOOL	TRUE: - In jog mode <i>Jog Axis</i> (HANDLING_KIN_JOG_AXIS): Jog in negative direction A1, A2, A3 or A4 - In jog mode <i>Jog Cartesian</i> (HANDLING_KIN_JOG_CART): Jog in negative direction X, Y, Z or A	I5':8 – 15 <i>Jog negative</i>
Variable: <i>Override</i> Data type: UINT			Correspondent in the PD interface
The variable for speed scaling in percent, for all operating modes except the referencing mode <i>Homing</i> (HANDLING_KIN_HOMING)			I5':Low-Byte <i>Override</i>
Variable: <i>ValueAdditional</i> Data type: LREAL			Correspondent in the PD interface
Reserved			–
Variable: <i>Diag</i> Data type: ST_AxisGroupHandlingKin_IN_Diag			Correspondent in the PD interface
The variable structure for activating the various diagnostic outputs, displayed in MessageHandler.			
<i>MessageAtState-Change</i>	BOOL	TRUE: The status information is output as information in MessageHandler for debugging purposes.	–
<i>MessageAtReadXml</i>	BOOL	TRUE: The errors that occur while reading the kinematics configuration XML files are output as information in MessageHandler for debugging purposes.	–

7.1.2 Status variables: AxisGroupHandlingKin.Inst[..].Out



15923990667

Data type: ST_AxisGroupHandlingKin_Out**Library:** 03_ApplicationModules\MPLCAxisGroupControl_HandlingKinematics.lib

Variable: Status Data type: ST_AxisGroupHandlingKin_Out_Status			Correspondent in the PD interface
The variable structure contains the following status variables that are independent of the operating mode.			
<i>AtLeastOneMotor-Turning</i>	BOOL	TRUE: At least 1 motor is turning (speed ≠ 0).	O1 ^r :0 <i>At least 1 motor is turning</i>
<i>AllInvertersReady</i>	BOOL	TRUE: The inverters are ready for interpolation/control by the controller.	O1 ^r :1 <i>All Frequency Inverters Ready</i>
<i>AllDrivesReferenced</i>	BOOL	TRUE: All drives are referenced. The signal is set to FALSE for at least 2 seconds after the increasing edge of the signal <i>Homing.Start[..]</i> .	O1 ^r :2 <i>All drives referenced</i>
<i>ApplicationReady</i>	BOOL	TRUE: The application is ready for operation.	O1 ^r :3 <i>Application ready</i>
<i>AllBrakesReleased</i>	BOOL	TRUE: The brakes of all drives are released.	O1 ^r :4 <i>All brakes released</i>
<i>WarningApplication</i>	BOOL	TRUE: If the motor speed is close to the maximum motor speed (90 – 100%), an application warning is generated.	O1 ^r :6 <i>Application warning</i>
<i>AtLeastOneInverter-Error</i>	BOOL	TRUE: At least one inverter is in error state.	O1 ^r :5 <i>At least 1 frequency inverter error</i>
<i>ErrorApplication</i>	BOOL	TRUE: Error of HandlingKinematics.	O1 ^r :7 <i>Application fault</i>
<i>AtLeastOneInverter-InSTO</i>	BOOL	TRUE: The safety function STO ¹⁾ was issued for at least 1 inverter. (<i>InverterData.SafeStop1/ SafeStop2</i> of <i>AxisInterface</i>).	O1 ^r :8 <i>At least 1 frequency inverter in "STO"</i>
<i>AllInvertersConnected</i>	BOOL	TRUE: The communication between controller and inverter has been established.	O1 ^r :9 <i>All frequency inverters connected</i>

Variable: Status Data type: ST_AxisGroupHandlingKin_Out_Status			Correspondent in the PD interface
<i>AllPowerAmplifiersEnabled</i>	BOOL	TRUE: The inverter output stages are enabled and deliver output voltage.	O1 ^r :10 <i>All output stages enabled</i>
<i>MaximalMotorSpeedCrossed</i>	BOOL	TRUE: The motor speed limit was exceeded.	O1 ^r :14 <i>Speed limit exceeded</i>
<i>WorkspaceLeft</i>	BOOL	TRUE: The work envelope of the kinematic model was left.	O1 ^r :15 <i>Work envelope left</i>

1) STO = SafeTorqueOff

Variable: Mode Data type: MC_AxisGroupHandlingKin_Mode		Correspondent in the PD interface
The variable shows the currently active operating mode.		O2 ^r :0 – 4 <i>Active operating mode</i>
- HANDLING_KIN_DEFAULT (Operating mode inactive <i>Default</i>) - HANDLING_KIN_JOG_AXIS (Axis-by-axis jog mode <i>Jog Axis</i>) - HANDLING_KIN_JOG_CART (Cartesian jog mode <i>Jog Cartesian</i>) - HANDLING_KIN_HOMING (Referencing mode <i>Homing</i>) - HANDLING_KIN_PROGRAM_AUTO (Program mode <i>Program Auto</i>) - HANDLING_KIN_PROGRAM_STEP (Program mode <i>Program Step</i>)		<i>Jog Axis</i> <i>Jog Cartesian</i> <i>Homing</i> <i>Program Auto</i> <i>Program Step</i>

Variable: Referenced Data type: ARRAY [1..8] OF BOOL		Correspondent in the PD interface
The variable shows the referenced axis. TRUE: The respective axis is referenced.		O2 ^r :8 – 15 <i>Axes referenced</i>

Variable: PrgNo Data type: UINT		Correspondent in the PD interface
The variable shows the number of the selected programs in program mode <i>Program Auto/Step</i> .		O3 ^r :High byte <i>Program number</i>

Variable: TelegramNo Data type: UINT		Correspondent in the PD interface
The variable shows the number of the selected telegram.		O3 ^r :Low-Byte <i>Telegram number</i>
INFORMATION: Only relevant, if HandlingKinematics is operated at an PLC via a fieldbus. In this case, the telegram number can be used to ensure the telegram consistency on the fieldbus: Only optional for process data profile 1, required for process data profiles 2 – 4. For details on consistent data transmission, refer to the manual "HandlingKinematics CCU application module".		

Variable: PrgStatus Data type: ST_AxisGroupHandlingKin_OUT_ProgramStatus			Correspondent in the PD interface
The variable structure indicates the status of the selected program. For detailed information refer to the manual "HandlingKinematics CCU application module".			
<i>Initialized</i>	BOOL	TRUE: The program is initialized.	O4 ^r :0 <i>Program initialized</i>
<i>BeingExecuted</i>	BOOL	TRUE: The program was restarted and is now run.	O4 ^r :1 <i>Program is executed</i>

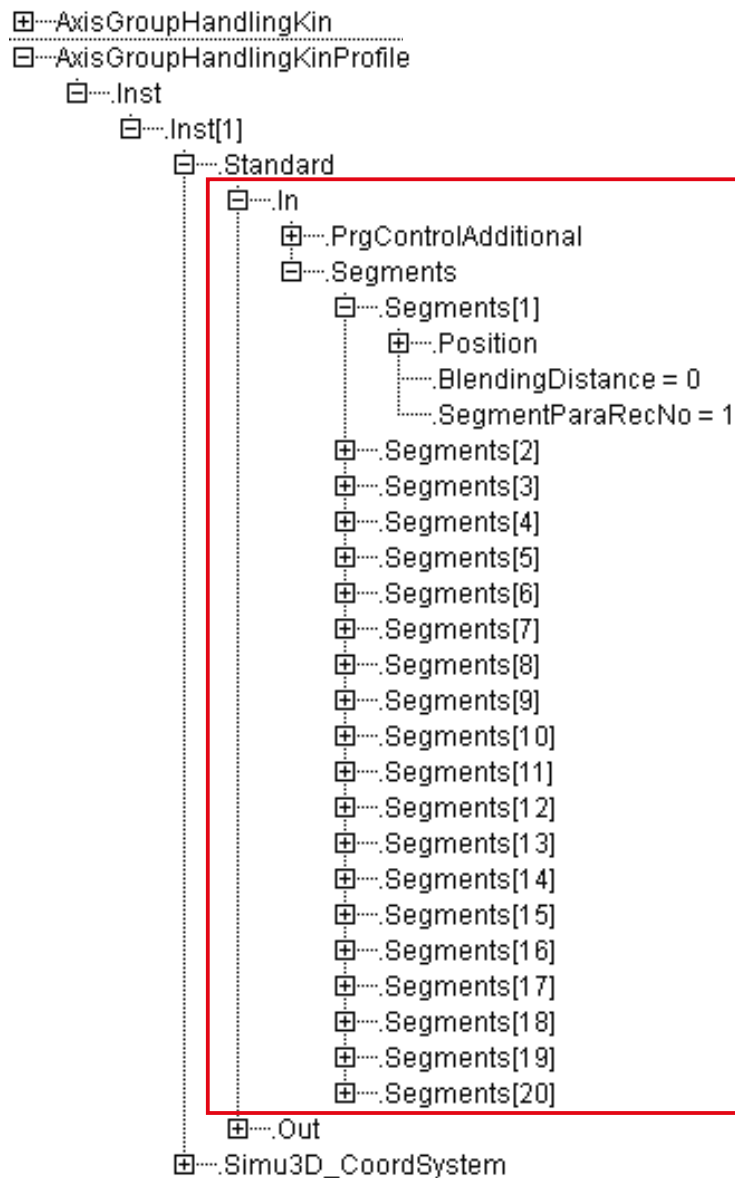
Variable: <i>PrgStatus</i> Data type: ST_AxisGroupHandlingKin_OUT_ProgramStatus			Correspondent in the PD interface
<i>Stopped</i>	BOOL	TRUE: The program was stopped while it was run.	O4':2 Program stopped
<i>Completed</i>	BOOL	TRUE: The program was run completely.	O4':3 Program finished
Variable: <i>ErrorID</i> Data type: DWORD			Correspondent in the PD interface
In case of an application error (<i>AxisGroupHandlingKin.Inst[..].Out.Status.ErrorApplication</i>), the error number of the error that first occurred is issued. For detailed information on error numbers, refer to the manual "HandlingKinematics CCU application module".			O5' ErrorID

7.2 Variable interface AxisGroupHandlingKinProfile.Inst[..]

The global variable structure *AxisGroupHandlingKinProfile* represents the application-specific part of the variable interface. The structures are contained as open source code in the "File types" tab of the PLC Editor project. When these structures are not changed, the behavior is identical to the "HandlingKinematics" application module. The structures can be extended to your requirements. Deleting elements from the structure is impossible, as it would lead to compiling errors.

Variable: <i>Inst</i> Data type: ARRAY [1..] OF ST_AxisGroupHandlingKinProfile_Instance	
The following variable structures are available for each kinematic instance.	
<i>Standard</i>	ST_AxisGroupHandlingKin_StandardProfile
<i>Simu3D_CoordSystem</i>	ST_Simu3D_CoordSystems

7.2.1 Control variables: AxisGroupHandlingKinProfile.Inst[..].Standard.In



15923993099

Data type: ST_AxisGroupHandlingKinProfile_In**Library:** 03_ApplicationModules\MPLCAxisGroupControl_HandlingKinematics.lib

Variable: <i>PrgControlAdditional</i> Data type: ST_AxisGroupHandlingKin_IN_StandardProf_Prog			Correspondent in the PD interface
The variable structure contains the following variables for controlling the program processing in program mode <i>Program Auto/Step</i> (HANDLING_KIN_PROGRAM_AUTO/HANDLING_KIN_PROGRAM_STEP).. For detailed information refer to the manual "HandlingKinematics CCU application module".			
<i>HandshakeFlagFor-SpecialProgs</i>	BOOL	Not used. Reserved for extensions.	–
<i>TouchprobeMotion</i>	BOOL	The value is adopted at program start. TRUE: Activates <i>Touchprobe Motion</i> (sensor-based positioning)	I4r:5 <i>Touchprobe Motion</i>

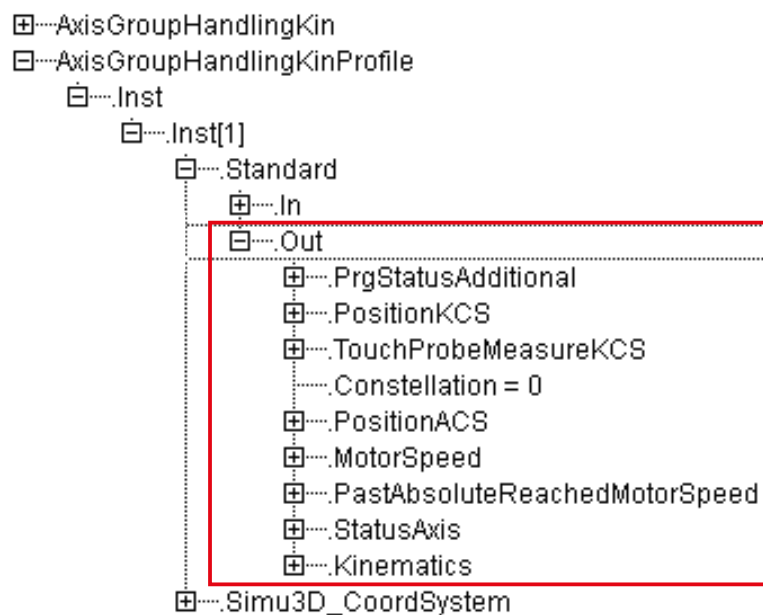
22323414/EN – 12/2015

Variable: <i>PrgControlAdditional</i> Data type: <i>ST_AxisGroupHandlingKin_IN_StandardProf_Prog</i>			Correspondent in the PD interface
<i>TouchprobeMeasure</i>	BOOL	The value is adopted at program start. TRUE: Activates <i>Touchprobe Measure</i> (position detection)	I4':6 <i>Touchprobe Measure</i>
<i>Wait1 – Wait20</i>	BOOL	The value is adopted at program start. TRUE: Forced stop at <i>Target pose 1 – 20</i> . FALSE: Traveling the next path segment was enabled.	I4':8 – 11 <i>Wait 1 – 4</i> I34':6 – 10 <i>Wait 5 – 9</i> I62':6 – 10 <i>Wait 10 – 14</i> I90':6 – 10 <i>Wait 15 – 19</i>
<i>End1 – End20</i>	BOOL	The value is adopted at program start. TRUE: Ends the program at <i>Target pose 1 – 20</i> , the first active end signal becomes valid.	I4':12 – 15 <i>End 1 – 4</i> I34':11 – 15 <i>End 5 – 9</i> I62':11 – 15 <i>End 10 – 14</i> I90':11 – 15 <i>End 15 – 19</i>
<i>RemainingDistance-Percentage</i>	UINT	Scaling of the remaining path length parameterized in the segment parameter sets in percent (<i>Touchprobe Motion</i>)	I6':High byte <i>Touchprobe: Remaining Distance Percentage</i>
<i>TelegramNo_060PD_STANDARD_PROFILE_10_SEGMENTS</i>	UINT	The telegram number ensures consistency in data transmission on the fieldbus. INFORMATION: Only relevant, if HandlingKinematics is operated at an PLC via a fieldbus. In this case, the telegram number must be used to ensure the telegram consistency on the fieldbus: Only optional for process data profile 1, required for process data profiles 2 – 4.	I33':Low-Byte <i>Telegram number</i>
<i>TelegramNo_088PD_STANDARD_PROFILE_15_SEGMENTS</i>	UINT	For details on consistent data transmission, refer to the manual "HandlingKinematics CCU application module".	I61':Low-Byte <i>Telegram number</i>
<i>TelegramNo_116PD_STANDARD_PROFILE_20_SEGMENTS</i>	UINT		I89':Low-Byte <i>Telegram number</i>

Variable: <i>Segments</i> Datentyp: ARRAY [1..20] OF <i>ST_AxisGroupHandlingKin_IN_StandardProf_Segments</i>	Correspondent in the PD interface
Array of path segment traveled by HandlingKinematics. A path segment is described by the following variables.	

Variable: Segments Datentyp: ARRAY [1..20] OF ST_AxisGroupHandlingKin_IN_StandardProf_Segments			Correspondent in the PD interface
<i>Position</i>	ARRAY [1..8] OF LREAL	<i>Target pose</i> specifications for the path segment: [1] = X, [2] = Y, [3] = Z, [4] = A, [5] = B, [6] = C, [7] = A7, [8] = A8	I8 ^r – I32 ^r I36 ^r – I60 ^r I64 ^r – I88 ^r I92 ^r – I116 ^r <i>Target pose 1 – 20 (XYZA) + Blending distance to segment 2 – 20</i>
<i>BlendingDistance</i>	LREAL	Distance to the <i>Target pose</i> of the previous path segment, from which is blended on the path segment currently observed.	
<i>SegmentParRecNo</i>	UINT	Selection of segment parameter set for the respective path segment	I7 ^r , I35 ^r , I63 ^r , I91 ^r Assignment of segment parameter sets to segments 1 – 5, 6 – 10, 11 – 15, 16 – 20

7.2.2 Status variables: AxisGroupHandlingKinProfile.Inst[..].Standard.Out



15923995531

Data type: ST_AxisGroupHandlingKinProfile_Out

Library: 03_ApplicationModules\MPLCAxisGroupControl_HandlingKinematics.lib

Variable: PrgStatusAdditional Data type: ST_AxisGroupHandlingKin_OUT_StandardProf_Prog			Correspondent in the PD interface
The variable structure contains the following feedback on the program processing in program mode <i>Program Auto/Step</i> (HANDLING_KIN_PROGRAM_AUTO/HANDLING_KIN_PROGRAM_STEP).. For detailed information refer to the manual "HandlingKinematics CCU application module".			
<i>TouchprobeMotion-Active</i>	BOOL	TRUE: <i>Touchprobe Motion</i> has been successfully activated for the current motion sequence.	O4 ^r :5 <i>Touchprobe Motion active</i>
<i>TouchprobeMotion-Detected</i>	BOOL	TRUE: The signal for <i>Touchprobe Motion</i> was successful recognized in the last path segment.	O4 ^r :6 <i>Touchprobe Motion recognized</i>

22323414/EN – 12/2015

Variable: <i>PrgStatusAdditional</i> Data type: ST_AxisGroupHandlingKin_OUT_StandardProf_Prog			Correspondent in the PD interface
<i>TouchprobeMeasureActive</i>	BOOL	TRUE: The signal for <i>Touchprobe Measure</i> has been successfully activated for the current motion sequence.	O4 ^r :7 <i>Touchprobe Measure active</i>
<i>WaitActive1 – WaitActive20</i>	BOOL	TRUE: The wait point at the respective <i>Target pose</i> is active. To continue the travel the signal <i>AxisGroupHandlingKinProfile.Inst[..].Standard.In.PrgControlAddiational.WaitX</i> must be set to FALSE.	O4 ^r :8 – 11 <i>Wait 1 – 4 active</i> O34 ^r :6 – 10 <i>Wait 5 – 9 active</i> O62 ^r :6 – 10 <i>Wait 10 – 14 active</i> O90 ^r :6 – 10 <i>Wait 15 – 19 active</i>
<i>EndActive1 – EndActive20</i>	BOOL	TRUE: The signal <i>AxisGroupHandlingKinProfile.Inst[..].Standard.In.PrgControlAddiational.EndX</i> moved the end of the motion sequence to the respective <i>Target pose</i> .	O4 ^r :12 – 15 <i>End 1 – 4 active</i> O34 ^r :11 – 15 <i>End 5 – 9 active</i> O62 ^r :11 – 15 <i>End 10 – 14 active</i> O90 ^r :11 – 15 <i>End 15 – 19 active</i>
<i>ActualSegmentID</i>	UINT	If the motor speed is close to the maximum motor speed (90 – 100%), an application warning is generated.	O32 ^r :High byte <i>Current SegmentID</i>
<i>TelegramNo_060PD_STANDARD_PROFILE_10_SEGMENTS</i>	UINT	The telegram number ensures consistency in data transmission on the fieldbus.	O33 ^r :Low-Byte <i>Telegram number</i>
<i>TelegramNo_088PD_STANDARD_PROFILE_15_SEGMENTS</i>	UINT	INFORMATION: The transferred telegram number is only relevant, if HandlingKinematics is operated at an PLC via a fieldbus. In this case, the telegram number must be used to ensure the telegram consistency on the fieldbus: Only optional for process data profile 1, required for process data profiles 2 – 4.	O61 ^r :Low-Byte <i>Telegram number</i>
<i>TelegramNo_116PD_STANDARD_PROFILE_20_SEGMENTS</i>	UINT	For details on consistent data transmission, refer to the manual "HandlingKinematics CCU application module".	O89 ^r :Low-Byte <i>Telegram number</i>
Variable: <i>PositionKCS</i> Data type: ARRAY [1..8] OF LREAL			Correspondent in the PD interface
Current Cartesian position in the kinematics coordinate system (KCS). <i>PositionKCS</i> or <i>TouchprobeMeasureKCS</i> are transferred via the fieldbus interface, controlled by the signal <i>AxisGroupHandlingKin.Inst[..].In.Control.DisplayOfTouchprobeMeasure</i> .			O7 ^r – O10 ^r <i>Position/Touchprobe Measure</i>

Variable: <i>TouchprobeMeasureKCS</i> Data type: ARRAY [1..8] OF LREAL	Correspondent in the PD interface
The Cartesian position (X, Y, Z coordinate) in the kinematics coordinate system (KCS), measured via the function <i>Touchprobe Measure. PostionKCS</i> or <i>Touchprobe-MeasureKCS</i> are transferred via the fieldbus interface, controlled by the signal <i>Axis-GroupHandlingKin.Inst[..].In.Control.DisplayOfTouchprobeMeasure</i> .	O7 ^r – O10 ^r <i>Position/Touchprobe Measure</i>
Variable: <i>Constellation</i> Data type: UINT	Correspondent in the PD interface
Current kinematics constellation	O11 ^r <i>Constellation</i>
Variable: <i>PositionACS</i> Data type: ARRAY [1..8] OF LREAL	Correspondent in the PD interface
Current position of axes 1 – 8 in user units	O12 ^r – O15 ^r <i>Axis value 1 – 4</i>
Variable: <i>MotorSpeed</i> Data type: ARRAY [1..8] OF LREAL	Correspondent in the PD interface
Current motor speed in 1/min. <i>MotorSpeed</i> or <i>PastAbsoluteReachedMotorSpeed</i> are transferred via the fieldbus interface, controlled by the signal <i>AxisGroupHandlingKin.Inst[..].In.Control.DisplayOfActMotorSpeed</i> .	O16 ^r – O19 ^r <i>Speed motor 1 – 4</i>
Variable: <i>PastAbsoluteReachedMotorSpeed</i> Data type: ARRAY [1..8] OF LREAL	Correspondent in the PD interface
Maximum motor speed in 1/min. <i>MotorSpeed</i> or <i>PastAbsoluteReachedMotorSpeed</i> are transferred via the fieldbus interface, controlled by the signal <i>AxisGroupHandlingKin.Inst[..].In.Control.DisplayOfActMotorSpeed</i> . The maximum motor speed is reset to 0 if the kinematics mode is changed or the edge of the signal <i>AxisGroupHandlingKin.Inst[..].In.Control.ErrorReset</i> rises.	O16 ^r – O19 ^r <i>Speed motor 1 – 4</i>

Variable: StatusAxis Data type: ARRAY [1..8] OF ST_AxisGroupHandlingKin_OUT_Standard-Prof_Axis			Correspondent in the PD interface
The variable structure contains axis granular status information.			O20 ^r – O25 ^r Status axis 1 – 6
<i>MotorTurning</i>	BOOL	TRUE: The motor turns (motor speed ≠ 0) provided that the inverter is connected (<i>AxisGroupHandlingKinProfile[..].Standard.Out.StatusAxis[..].InverterConnected</i>) (<i>InverterData.MotorStandstill</i> of <i>AxisInterface</i>).	
<i>InverterReady</i>	BOOL	TRUE: The inverter is ready for interpolation/control by the controller (<i>InverterData.InverterReady</i> and no <i>FBEError</i> of <i>AxisInterface</i>).	
<i>DriveReferenced</i>	BOOL	TRUE: The drive is now referenced. The signal is set to FALSE for at least 2 seconds after the increasing edge of the signal <i>AxisGroupHandlingKin[..].In.Homing.Start[..]</i> (<i>InverterData.Referenced</i> of <i>AxisInterface</i>).	
<i>PowerAmplifier Enabled</i>	BOOL	TRUE: The output stage of the inverter is enabled and provides the output voltage (<i>Inverter.Powered</i> of <i>AxisInterface</i>).	
<i>BrakeReleased</i>	BOOL	TRUE: The brake of the drive / the axis is released (<i>Inverter.BrakeReleased</i> of <i>AxisInterface</i>).	
<i>InverterConnected</i>	BOOL	TRUE: The communication between controller and inverter is established (<i>Connected</i> of <i>AxisInterface</i>).	
<i>InverterError</i>	BOOL	TRUE: Fault on the inverter.	
<i>InverterStatus</i>	BYTE	Status number or inverter fault number (<i>InverterData.FaultStatus/InverterStatus</i> of <i>AxisInterface</i>).	

Variable: Kinematics Data type: ST_AxisGroupHandlingKin_OUT_StandardProf_Kin			Correspondent in the PD interface
The variable structure contains status information on the state of the kinematic control.			
<i>ActualMotionCommandCompleted</i>	BOOL	TRUE: The current motion command is completed. The current path segment is traveled. This can have the following 2 reasons: - When <i>AxisGroupHandlingKin.Inst[..].Out.PrgStatus.Completed</i> = FALSE: The kinematics has stopped at a wait point. - When <i>AxisGroupHandlingKin.Inst[..].Out.PrgStatus.Completed</i> = TRUE: The kinematics has completed traveling the motion sequence. SRL program is completed.	O26 ^r :0 Current Motion Task Complete
<i>CpPathEndReached</i>	BOOL	Only for program 3 and 4. TRUE: The path end of the path interpolation (CP) is reached.	O26 ^r :1 CP Path End Reached

Variable: <i>Kinematics</i> Data type: ST_AxisGroupHandlingKin_OUT_StandardProf_Kin			Correspondent in the PD interface
<i>BackToPathActive</i>	BOOL	Only for program 3 and 4. TRUE: The path was left. About to repositioning or repositioning currently performed (BackTo-Path). When the repositioning is completed, the signal changes back to FALSE.	O26 ^f :2 <i>Repositioning to CP Path Active</i>
<i>CpMotionSlowed-Down</i>	BOOL	Only for program 3 and 4. TRUE: The look ahead function has automatically decelerated the specified movement, to not breach any speed or acceleration limits. The signal is reset to FALSE when a trajectory is set that must not be decelerated.	O26 ^f :3 <i>Specified CP Motion Decelerated</i>
<i>CartesianPosition-Valid</i>	BOOL	TRUE: The issued position, orientation and constellation in <i>AxisGroupHandlingKinProfile.Inst[..].Standard.Out.PositionKCS</i> and <i>.Constellation</i> are valid. The kinematics configuration is complete and correctly calculates the transformations. FALSE: When this bit is set to FALSE, position, orientation, and constellation must not be used in the PLC.	O26 ^f :4 <i>Cartesian Position Valid</i>
<i>KinMode</i>	BOOL	Operating mode of the lower-level "Kinematics" technology module. Depending on the selected HandlingKinematics mode: • Default: 0 = KIN_AM_DEFAULT • Axis-by-axis jog mode: 110 = KIN_JOG_AXIS • Cartesian jog mode: 111 = KIN_JOG_CART • Referencing: 2 = KIN_AM_HOMING • Program auto/step Program 1: 120 = KIN_TARGET_AXIS • Program auto/step Program 2: 121=KIN_TARGET_CART • Program auto/step Program 3 and 4: – 213 = KIN_LIN_3D – 300 = KIN_CP_BLENDING – KIN_TARGET_AXIS (only for BackToPath) • Program auto/step TouchprobeMotion: – 121 = KIN_TARGET_CART Other operating modes exist, but are not relevant for HandlingKinematics. For detailed information, refer to chapter "Interpolate operating modes" in the manual "Kinematics technology module for MultiMotion/MultiMotion Light".	O27 ^f <i>Kinematics mode</i>
<i>TranslationVel</i>	LREAL	Effective translational velocity of the tool center point (TCP) that contains the speed in direction X, Y, and Z.	O28 ^f <i>Translational velocity</i>

Variable: <i>Kinematics</i> Data type: ST_AxisGroupHandlingKin_OUT_StandardProf_Kin			Correspondent in the PD interface
<i>PathTargetDistance</i>	LREAL	Only for program 3 and 4: The remaining segment on the path to the <i>Target pose</i> , where the next wait point is active or the path ends.	O31' <i>CP Path: Target distance</i>
<i>RemainingSegments</i>	BOOL	Only for program 3 and 4: The number of CP path segments to the <i>Target pose</i> , where the next wait point is active or the path ends.	O26'3 <i>Remaining CP Segments</i>

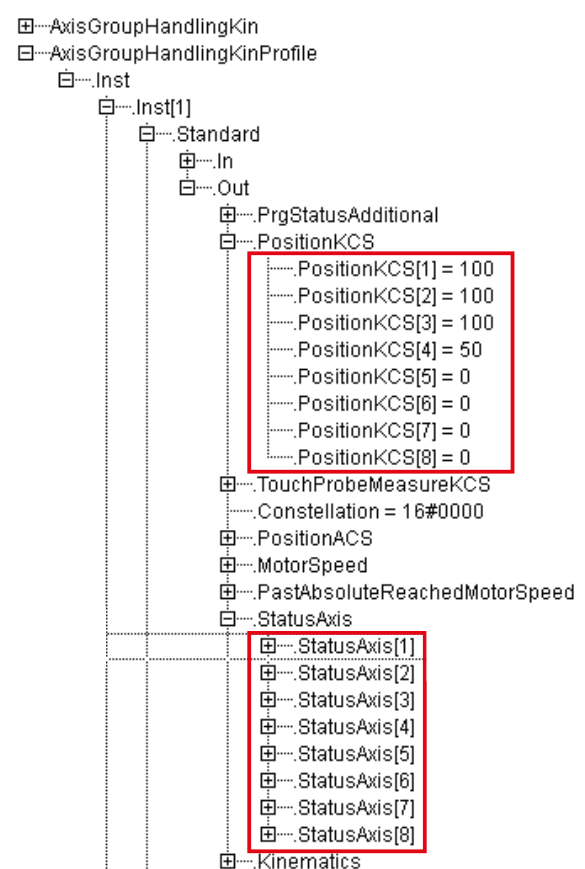
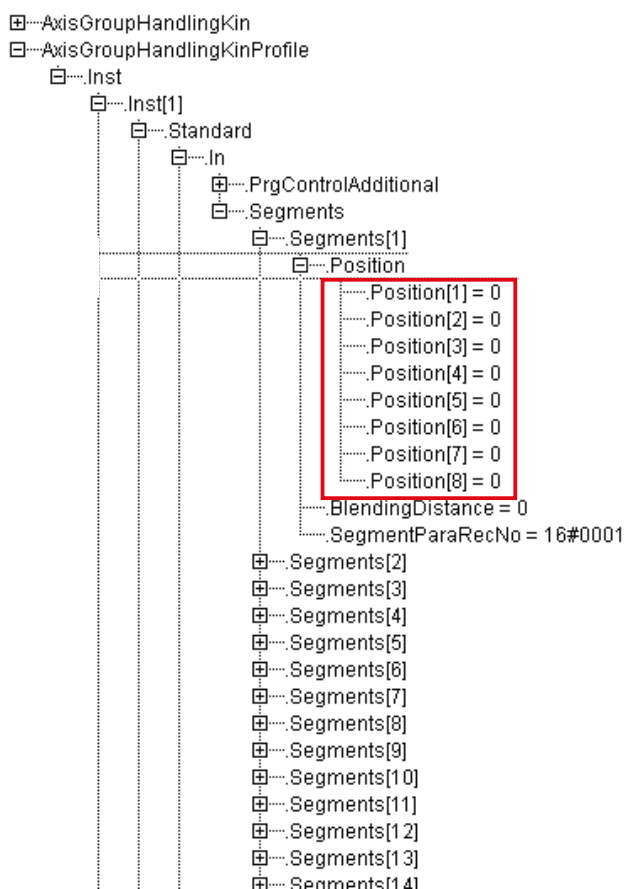
8 Applications

8.1 Controlling the B/C degree of freedom and the A7/A8 auxiliary axes

The "HandlingKinematics" CCU application module is limited to kinematic models with the Cartesian degrees of freedom X, Y, Z and A (rotation around the Z axis).

This limitation is removed in the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module. Kinematic models with Cartesian degrees of freedom B (rotation around the Y axis) and C (rotation around the X axis) can also be used here. In addition, you can control the kinematic auxiliary axes A7/A8.

The variable interface *AxisGroupHandlingKinProfile* allows for programming all 6 degrees of freedom/kinematic axes and the kinematic auxiliary axes A7/A8. The segment variables are performed as array [1 – 8] so that the target poses of all degrees of freedom and auxiliary axes can be defined in each path segment. The status information of all axes are output.



15925380363

During the configuration, the kinematic models with the degrees of freedom B/C, and the auxiliary axes A7/A8 can be selected as follows:

- From MotionStudio® authorization level "Advanced" (input: 5249) or
- via the "Advanced Configuration"

The configuration of the motion parameters in the segment parameter sets for the degrees of freedom B/C, and for the auxiliary axes A7/A8, is performed via the parameter list that is displayed at the end of the configuration.

In the various SRL programs are limits for using the degrees of freedom B/C and the auxiliary axes A7/A8, as shown in the following table.

SRL programs	Degrees of freedom B/C	Auxiliary axes A7/A8
Program 1	Supported	Supported with limits The target pose is reached by A7/A8. It is not checked if the auxiliary axes reached the target poses when switching to the next path segment.
Program 2	Supported	
Program 3	Not supported The degrees of freedom B/C not moved.	Not supported The auxiliary axes A7/A8 not moved.
Program 4	Supported	Supported

Only program 4 supports all degrees of freedom and auxiliary axes without limitation. This program is recommended for using the degrees of freedom B/C and auxiliary axes A7/A8.

The kinematic constellation can be changed in program 1, even when the auxiliary axes are used. In this case, a program consisting of 1 path segment is traveled, while only the constellation is changed and the auxiliary axes are not moved. Then, the paths can be traveled in the changed constellation using program 4.

To control the technology module via fieldbus, the process data interface must be adjusted as described in the following chapter.

8.2 Changing the existing fieldbus mapping or integrating a user fieldbus mapping

You can either change the fieldbus mapping in the existing actions `STANDARD_PROFILE_32PD` and `STANDARD_PROFILE_PLUS_28PD` or integrate your own fieldbus mapping.

Changing the existing fieldbus mapping

As standard, the actions `STANDARD_PROFILE_32PD` and `STANDARD_PROFILE_PLUS_28PD` are contained in the following programs as open source code:

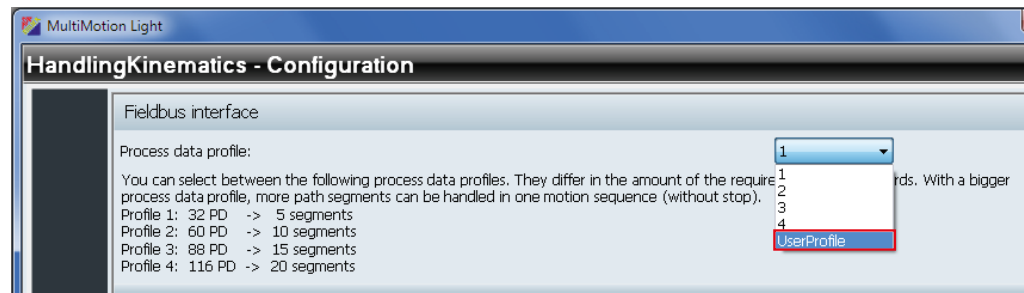
- `MC_FieldbusMappingIN_HandlingKin_StandardProf`
- `MC_FieldbusMappingOUT_HandlingKin_StandardProf`

The programs and actions are described in the following chapters.

The fieldbus mapping can for example be extended at a free slot in the fieldbus interface, or replace values that are not required by values that are relevant for the application.

Integrating the user fieldbus mapping

A fieldbus mapping of your choice can be integrated in the action `USER_SPECIFIC_PROFILE` that is empty as standard. For example, the source code from existing actions can be used and adapted in the user profile. When configuring the technology module, you can select the user profile to be automatically opened.

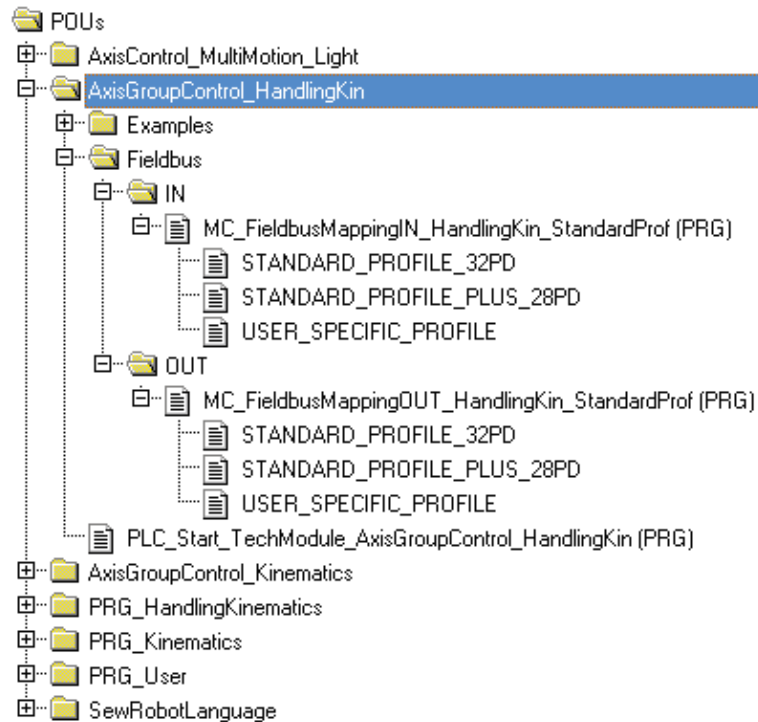


16094691083

Fieldbus mapping	Advantages	Disadvantages
Changing	Small changes to the fieldbus interface are possible quickly.	The standard behavior of HandlingKinematics with fieldbus control is no longer possible. If you want to return to the standard mapping of HandlingKinematics, as in the "HandlingKinematics" CCU application module, the original source code must be integrated.
Integrating	The existing actions remain unchanged and can still be selected during configuration.	Higher effort implementing. When the user profile is longer than 32 PD, ensure consistency.

8.2.1 Fieldbus mapping programs

The programs for fieldbus mapping are available as open source code.

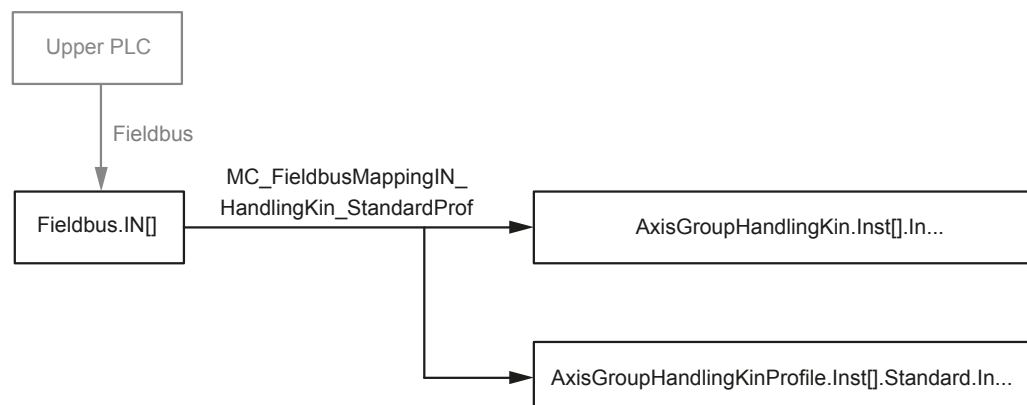


16097766027

MC_FieldbusMappingIN_HandlingKin_StandardProf

The fieldbus input data (SPS → controller MOVI-PLC®) is mapped to the IN structure of the variable interface (assigned and written) in the program `MC_FieldbusMappingIN_HandlingKin_StandardProf`.

Information is copied from the structure `Fieldbus.IN` (ARRAY[1 – 120] OF WORD) to the variable interfaces `AxisGroupHandlingKin.Inst[n].In` and `AxisGroupHandlingKinProfile.Inst[n].Standard.In`. The input data transferred by the fieldbus is automatically stored in the structure `Fieldbus.IN`.

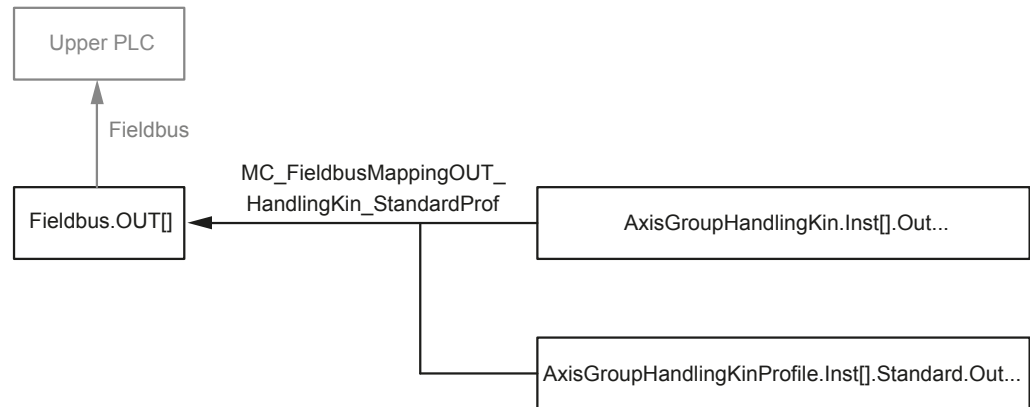


16095374475

MC_FieldbusMappingOUT_HandlingKin_StandardProf

The fieldbus output data (controller MOVI-PLC® → SPS) is mapped to the OUT structure of the variable interface (assigned and written) in the program `MC_FieldbusMappingOUT_HandlingKin_StandardProf`.

Information is copied from the variable interfaces *AxisGroupHandlingKin.Inst[n].Out* and *AxisGroupHandlingKinProfile.Inst[n].Standard.Out* to the structure `Fieldbus.OUT` (ARRAY[1..120] OF WORD).



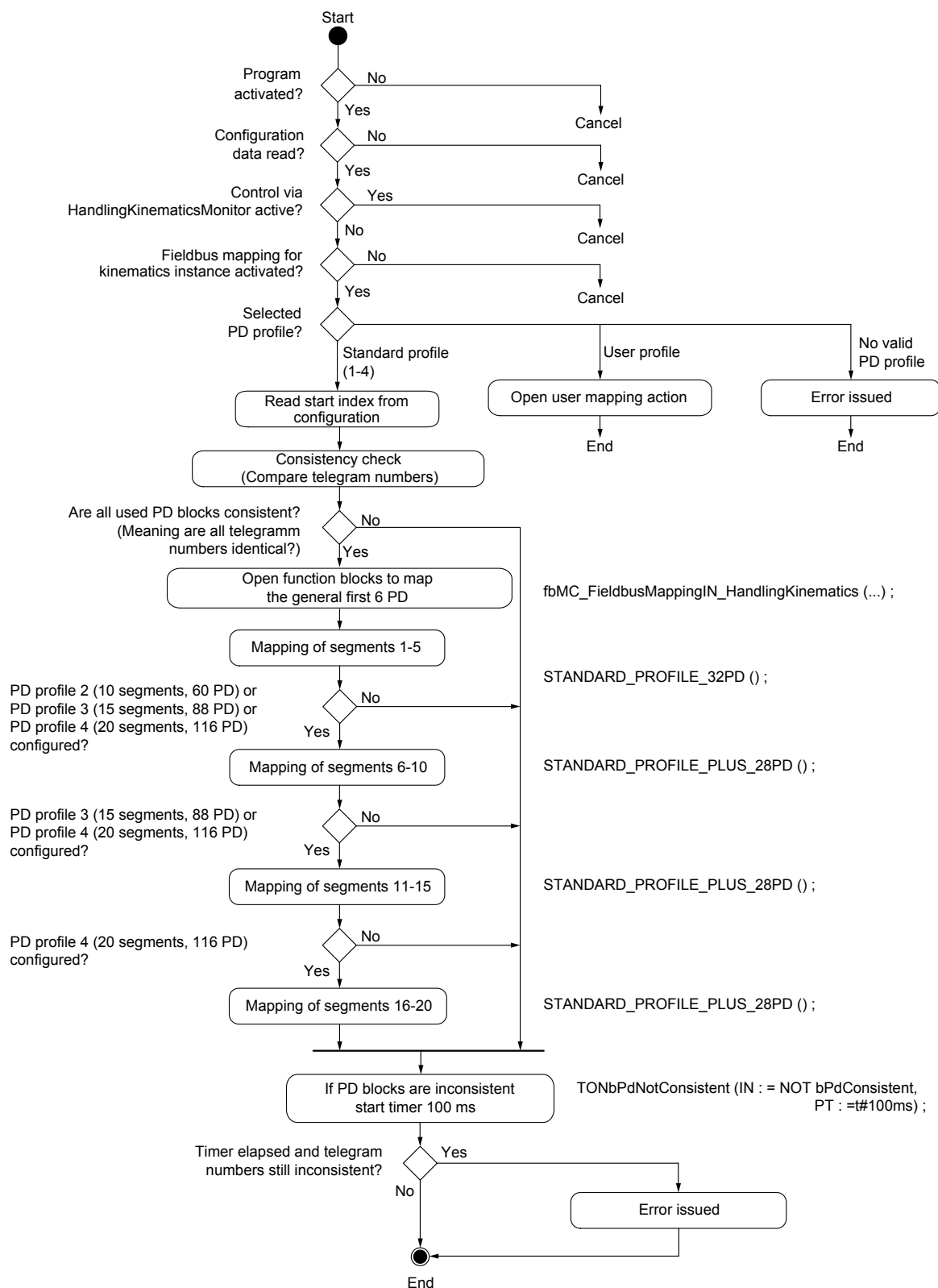
16095376907

8.2.2 Flow diagrams of fieldbus mapping programs

The fieldbus mapping programs contain actions that map the variables.

MC_FieldbusMappingIN_HandlingKin_StandardProf

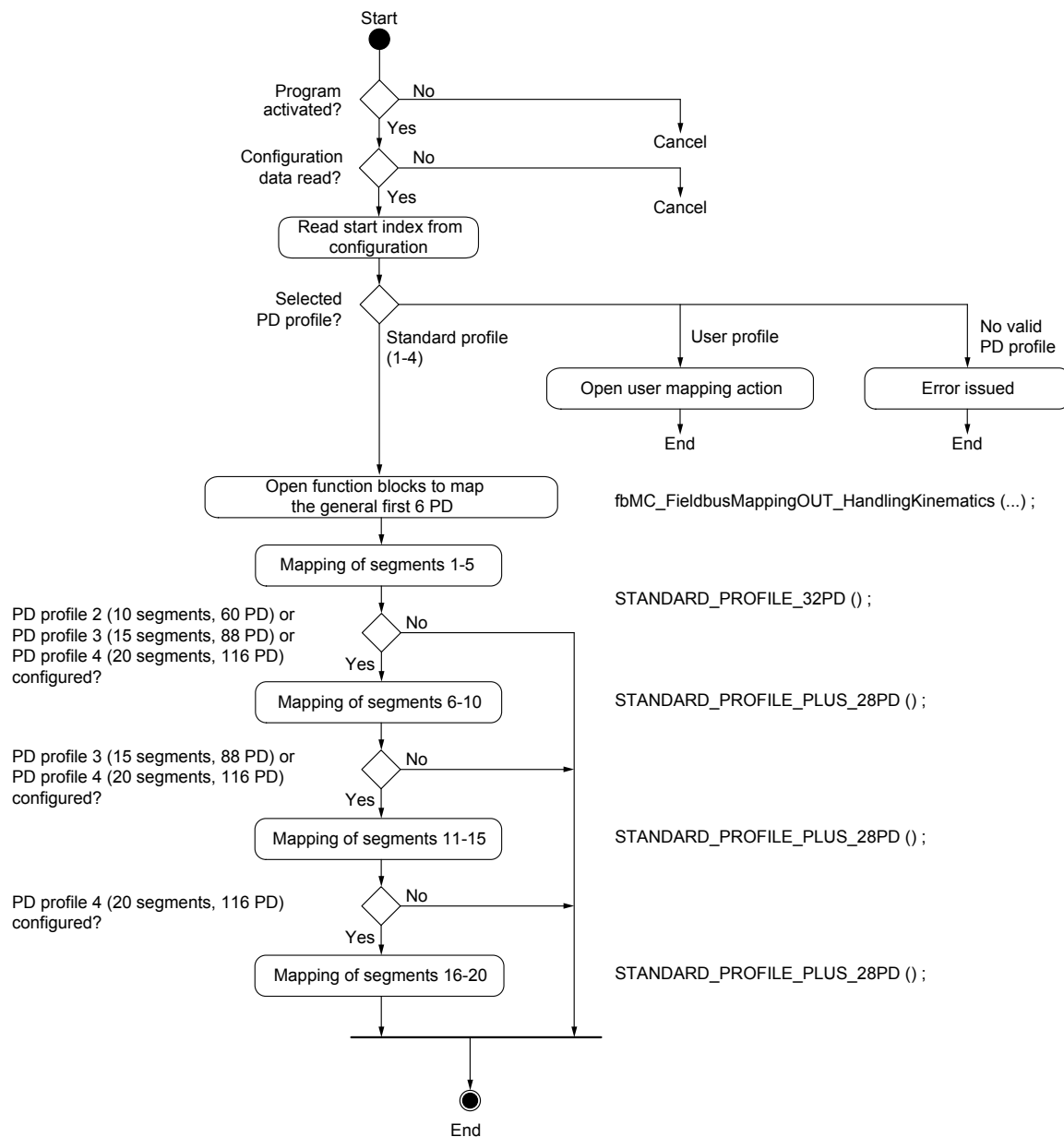
The actions for fieldbus mapping are opened in the program `MC_FieldbusMappingIN_HandlingKin_StandardProf` according to the following procedure.



16097760651

MC_FieldbusMappingOUT_HandlingKin_StandardProf

The actions for fieldbus mapping are opened in the program `MC_FieldbusMappingOUT_HandlingKin_StandardProf` according to the following procedure.



16097763083

8.2.3 Adjusting the templates in the process data monitor

The templates (→ 34) loaded in the process data monitor can be adjusted to your requirements.

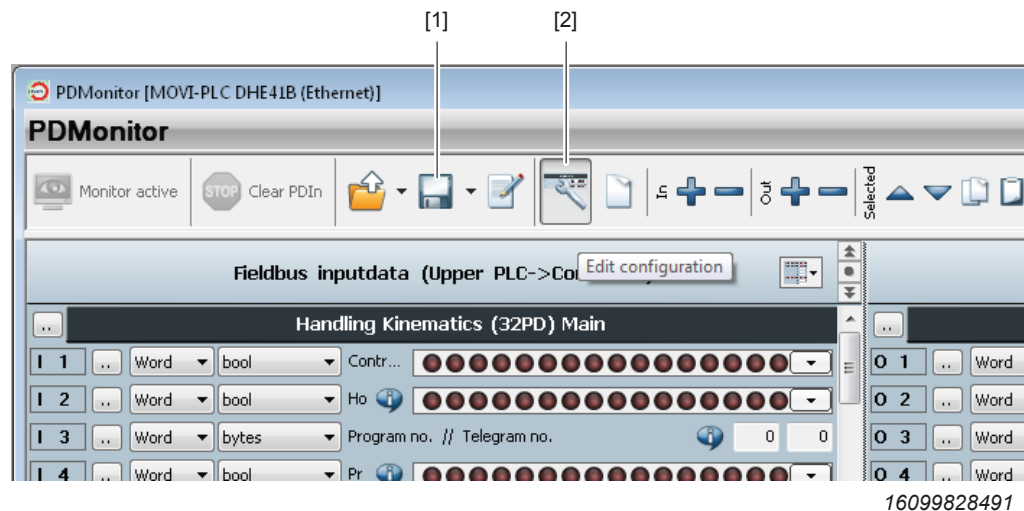
When the fieldbus mapping is changed in the control program and the process data monitor is still to be used, the changes must also be performed in the process data monitor.

Adjusting a template

Proceed as follows:

✓ The process data monitor is opened and a template is loaded.

1. To switch to the edit mode, click the button [2].



2. Adjust the template to your requirements.
 3. To save the settings, click the button [1].
 4. Save the template under a new name.
- ⇒ The template is adjusted.

8.3 Integrating a customized kinematics model

To include a customized kinematic model, the library `MPLCKinematics_UserKin.lib` must be replaced. Contact SEW-EURODRIVE in such cases.

8.4 Simultaneous use of Kinematics and HandlingKinematics

When the technology modules "Kinematics" and "HandlingKinematics for MultiMotion/MultiMotion Light" are run simultaneously with a MOVI-PLC® power controller, only the technology module "HandlingKinematics for MultiMotion/MultiMotion Light" (→ 23) is integrated. While configuring the relevant kinematic instance, select if "Kinematics" or "HandlingKinematics" (→ 29) are to be used for this kinematic instance.

Index

Numerical

3D simulation..... 16, 18

A

Application programming..... 39
 AxisGroupHandlingKin.Inst[.].In 43
 AxisGroupHandlingKin.Inst[.].Out..... 47
 AxisGroupHandlingKinProfile.Inst[.].Standard.In 50
 AxisGroupHandlingKinProfile.Inst[.].Standard.Out.. 52

B

Bus system 9

C

Configuration
 Create new 29
 Starting 29
 Control variables 38, 43, 50
 Copyright notice 7

D

Diagnostics tools
 3D simulation..... 18
 HandlingKinematicsMonitor..... 18
 MessageHandler 18
 Documents, applicable 7
 Drive startup for MOVI-PLC® 18, 28

E

Embedded safety notes..... 6
 Exclusion of liability 6

F

Fieldbus mapping
 Activating..... 33
 Changing 60
 Deactivating..... 33
 Integrating 60
 Set start index 33
 Folder "TechData"
 Copying via card reader 26
 Functions 15

H

HandlingKinematicsMonitor..... 16, 18
 Hardware 11
 Hazard symbols
 Meaning..... 5

K

Kinematics Configurator 16, 18

L

Liability 6
 Liability for defects..... 6

M

Macros..... 23
 MessageHandler 16, 18
 MotionStudio® project 20
 MOVITOOLS® MotionStudio
 Handling 20
 Scope of delivery 18

N

Notes
 Designation in the documentation 5
 Meaning of the hazard symbols 5

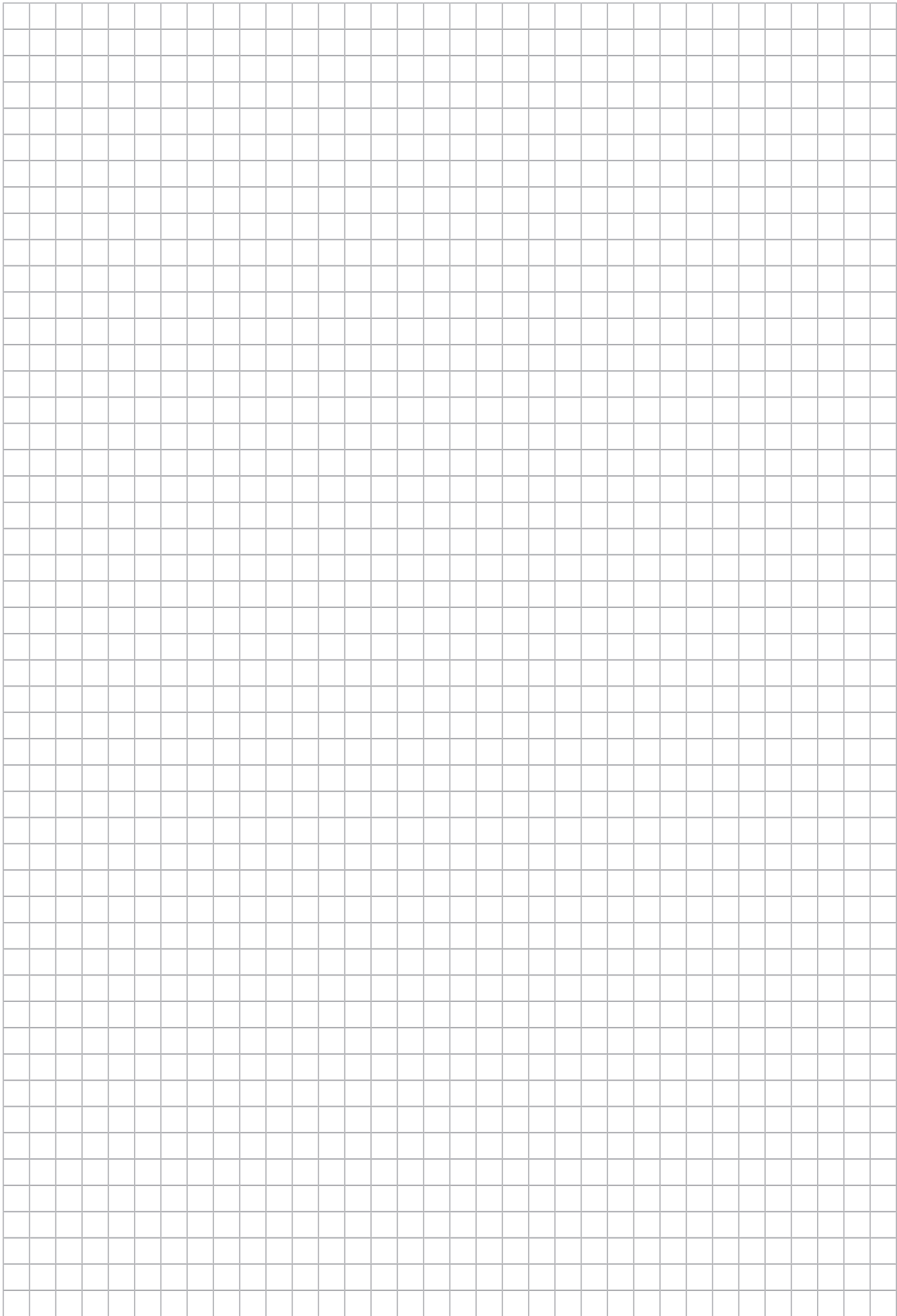
O

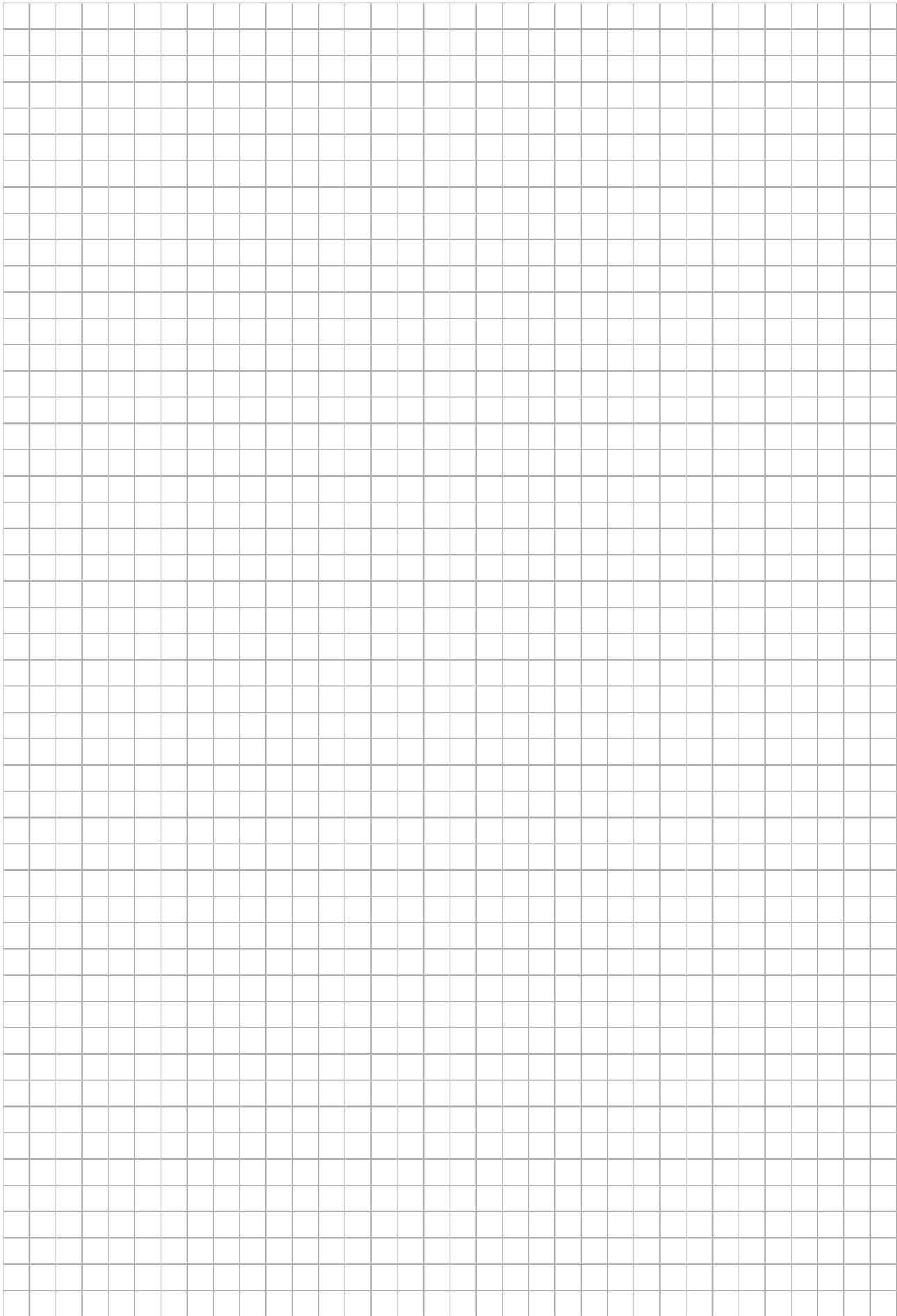
Operating modes..... 14
 Other applicable documentation..... 7

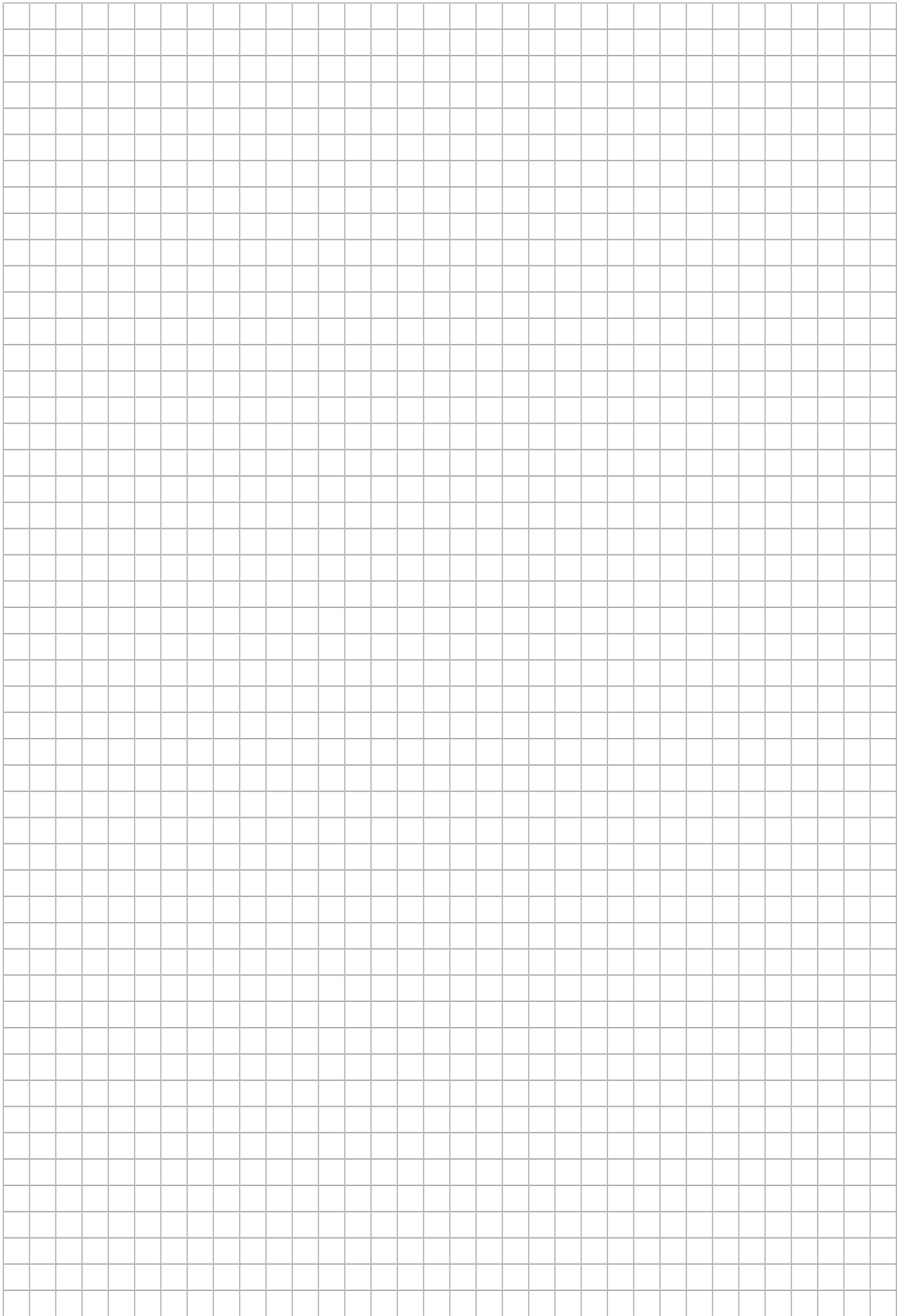
P

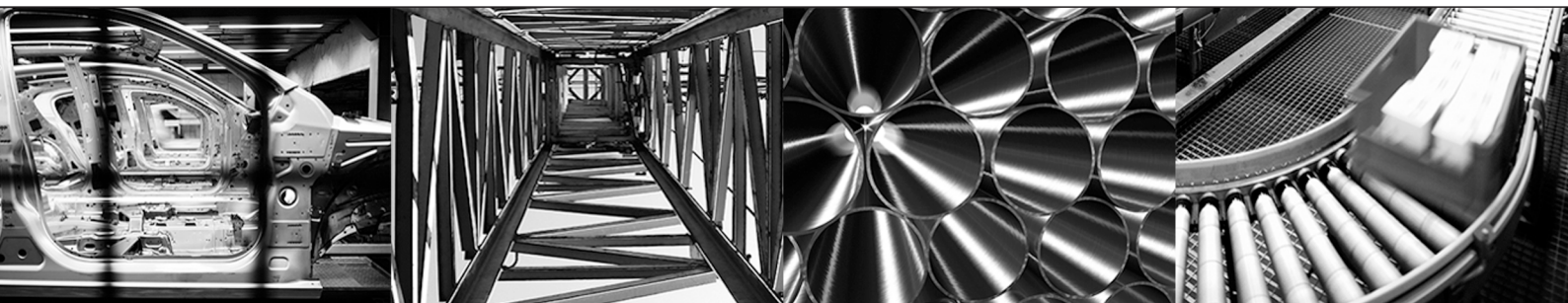
Pick and place programs..... 36
 PLC Editor 18
 PLC Editor project 22
 Process data monitor 16
 Scope of functions..... 34
 Selecting a template..... 34
 Starting 34
 Product names 7
 Project planning notes
 Hardware..... 11
 Software 11
 Technology points 11
 Project templates
 AxisControl_MultiMotion.pro 22

AxisControl_MultiMotion_Light.pro.....	22	Task configuration	
R		Adding task entries.....	32
Requirements		Creating.....	32
Configuration	11, 18	Expanding	32
S		Integrating	25
Safety Notes	8	Maintained	25
Designation in the documentation	5	Technology module	
General.....	8	Adding	30
Meaning of the hazard symbols	5	Configure.....	31
Structure of embedded.....	6	Integrating using macros	23
Structure of the section-related	5	Manually integrating	24
Scope of delivery	18	Operating.....	36
Section-related safety notes.....	5	Technology points	11
Sequential program	38	Templates.....	35
Setting		Tools	
Baud rate	28	3D simulation.....	16
SBus address	28	HandlingKinematicsMonitor.....	16
Signal words in the safety notes.....	5	Kinematics Configurator	16
Software	11	MessageHandler	16
SRL files	26	Process data monitor	16
Startup		Trademarks	7
Adding technology module	29	U	
Copying SRL files.....	26	Use	
Creating a MotionStudio® project	20	designated	9
Creating a PLC Editor project.....	22	with PLC	14
Integrating the technology module	23	without PLC	14
Procedure.....	19	User kinematic library	24
Starting up the single axes	28	User kinematic model	24
Status variables.....	47, 52	V	
T		Variable interface	36, 42, 49
Target group	8	Variables	
		AxisGroupHandlingKin	42
		AxisGroupHandlingKinProfile	49











SEW-EURODRIVE
Driving the world

SEW
EURODRIVE

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023
76642 BRUCHSAL
GERMANY
Phone +49 7251 75-0
Fax +49 7251 75-1970
sew@sew-eurodrive.com
→ www.sew-eurodrive.com