BECKHOFF New Automation Technology

Manual | EN

TS1110

TwinCAT 2 | Simulation Manager

Supplement | System

2021-11-11 | Version: 1.0

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Notes on information security

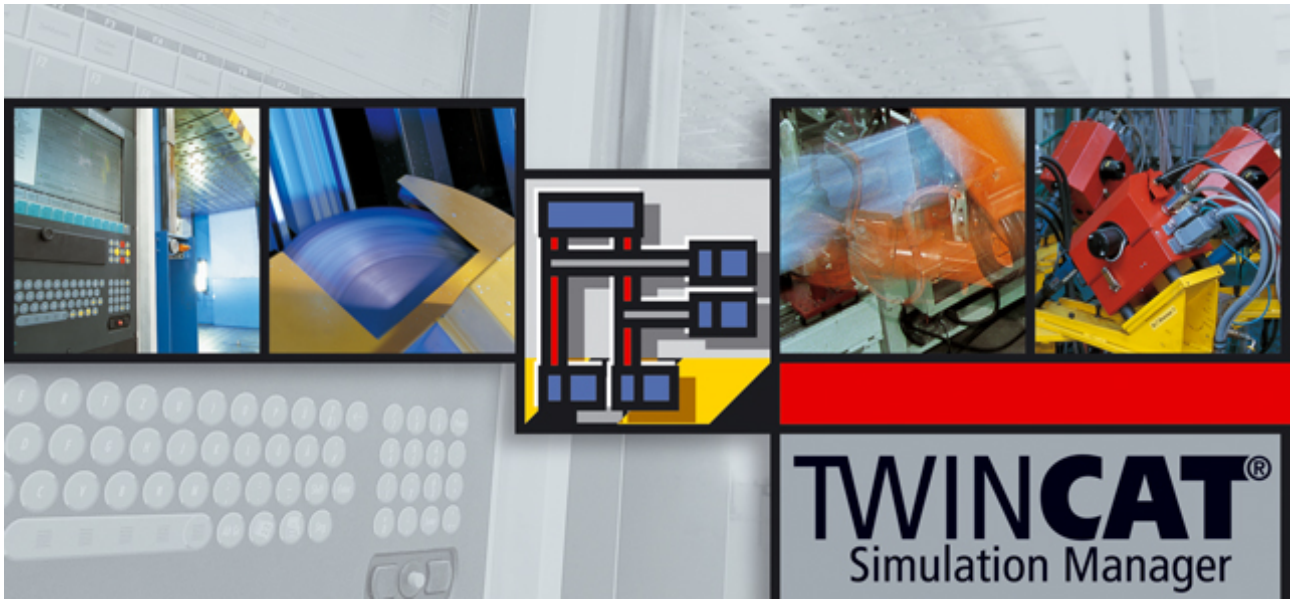
The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 TwinCAT Simulation Manager



The '**TwinCAT Simulation Manager**' is a powerful tool for simulating machines or machine systems.

There is a requirement in various scenarios for the partial or complete simulation of a production machine or plant:

- Development
  Hardware and software are developed in parallel, usually by different people, or often become available at different times. A simulation environment is indispensable for testing the plant/machine in sections. The '**TwinCAT Simulation Manager**' supports the creation of a "virtual machine", which corresponds to the real machine in its performance characteristics. Due to the integration of the simulation in the real-time system, even real-time-critical requirements can be realised (real-time simulation).
  --> The dependency of the software on the hardware is reduced. It is even possible to complete the software ahead of actual commissioning. The quality of the machine software is increased by developer tests.
- Acceptance of software/machine before commissioning
  The '**TwinCAT Simulation Manager**' enables the implementation of acceptance rules in the simulation. This may be the normal operating performance of the machine / plant, or the testing of compliance with interface specifications or the simulation of faults. Those responsible for quality assurance are now armed with a powerful tool.
  --> Assurance of quality levels
- Commissioning the machine / plant
  The commissioning of individual machine sections is enabled. Through advance tests (see Acceptance of machine/software before commissioning), more attention can be devoted to the parameterisation and optimisation of the machine.
  --> The time required for commissioning is considerably reduced.
- In productions
  Machines / plant sections break down; material bottlenecks may arise. The '**TwinCAT Simulation Manager**' is able to maintain the partial operation of the machine / plant through various profiles.
  --> The machine / plant can be used more flexibly.
- Education and Demonstration
  The simulation of the hardware allows to execute the machine software purely virtual on a demo system which can be used to train the machine operator or service technician. Additionally suits the simulation system for demo purposes in different scenarios.
  --> Supports the education of machine / plant users and service staff.

All these scenarios are supported by the TwinCAT Simulation Manager, which places an extensive range of tools for the creation and administration of simulation environments at the fingertips or the developer / plant operator:

- Administration and configuration of simulation environments / profiles
- Simulation of I/O devices and axes
- Simple and secure switchover into simulation mode
- "Virtual device" and "virtual function" libraries
- Automatic creation of a real-time capable simulation PLC project framework with symbols and commentaries taken over from the original project
- Use of all the options of the PLC programming environment, also in simulation code: e.g. graphic programming, tracing, debugging etc.

Principle of operation

I/O devices and axes are initially connected to the PLC. If parts of the machine are to be simulated, then the devices in question must be deactivated. The performance of the relevant inputs and outputs must be replicated as exact as possible so that the machine PLC – henceforth referred to as the original PLC – can continue to perform its functions. This is carried out by the simulation PLC, which connects itself to the system in place of the inactive I/O devices.

For coupling, the mapping from the original PLC to the inactive device must be broken and rerouted to the simulation PLC. Deactivated axes are replicated by a simulation axis; any latch functionalities in the simulation PLC are implemented. Simulated axes are switched over to 'free running' in the '**TwinCAT Simulation Manager**'.

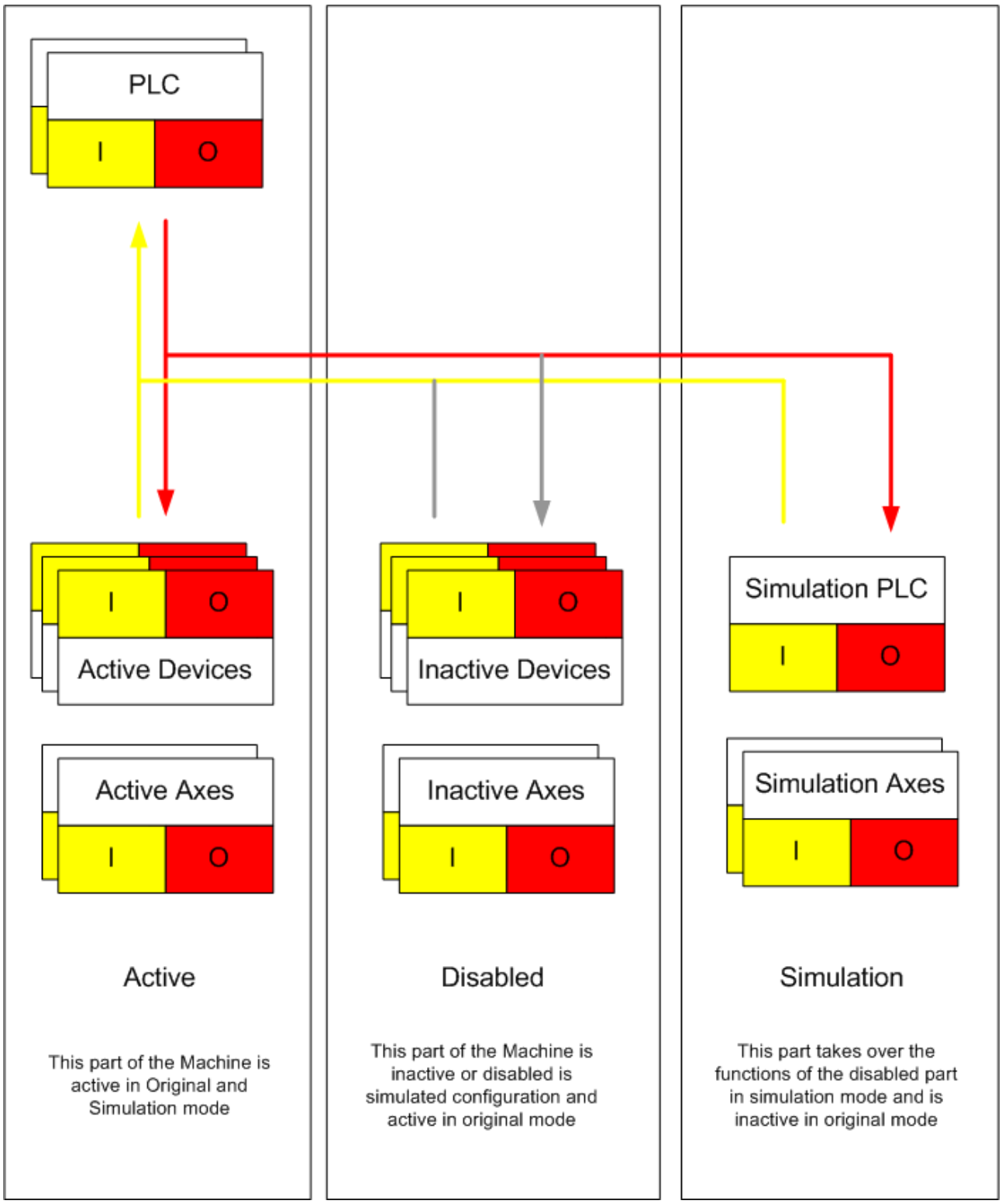


Fig. 1: Figure 1: Principle of the TwinCAT Simulation Manager

The replication of the I/O functionality in the simulation PLC is simplified by a set of simulation blocks. Complex performance can be easily replicated by the mapping of these blocks and encapsulated in the POUs or reused.

TwinCAT realizes the data transfer between PLC and I/O typically via IO-Mappings. The TwinCAT System Manager creates these mappings and initializes the adequate memory copy instructions within the TwinCAT Runtime system (task-synchronized memory copy between process-images). This sort of mapping is only useable within a local scenario, which means within one single Resource / TwinCAT System or CPU where in-resource memory access to the process images is available.

In a distributed system environment - where several resources / CPUs are involved a different technology for the data exchange must be used. This could be realized via discrete wireing, a fieldbus or common network technology.

To support distributed as well as local scenarios the **TwinCAT Simulation Manager** uses internally an abstract mapping model that transparently hides the data-exchange protocol. The **TwinCAT Simulation Manager** chooses the appropriate protocol dependent on the location of the communication start- and endpoint.

Actually, the following protocols are supported:

- TwinCAT IO-Mappings for local in-ressource links / mappings (highest data throughput, limited by the cycle of the used task-master)
- Network Publisher-Subscriber Variables for remote links (EtherCAT Master-Master protocol, data througput limited by EtherCAT protocol and RT-Ethernet cycle)

By the transparent usage of remote devices, it is possible to deploy and execute the machine software (original software) and the simulation software freely on the available resources/CPUs. The system smoothly scales from local to highly distributed scenarios.

2.1 Version History

2.1.1 New in Version 2.1

Previous versions of the TwinCAT Simulation Manager (< version 2.1) preassumed an available executeable system environment. Existing hardware, activated System Manager configurations and activated/running PLC programs were therefore used as starting point for a TwinCAT Simulation Manager project. The version 2.1 now offers features for distribution and deployment of simulation projects and configurations. The IDE of the TwinCAT Simulation Manager enables to assign the target systems (as Original- or Simulation-Targets), to select and to activate available System Manager configurations and to deploy PLC-Programs on available CPU resources.

These features will be summerized under the generic term "Deployment" and include:

- The import/export of preexisting System Manager Configurations and the assignment of target systems in a distributed environment.
- The import/ export, creation and deletion of PLC Projects in the context of the Simulation Manager project (with allocation/deallocation of PLC-Runtimes)
- The creation of bootprojects directly out of the TwinCAT Simulation Manager IDE.

These new product abilities simplify to transfer a preexisting and executable machine environment into a simulation-, demo- or education-environment. Local, mixed and distributed scenarios will be supported.

Furthermore the following extensions and enhancements are implemented:

- Support of CNC axes (usage of the CNC Simulation mode, CNC manual functions)
- Support of TwinCAT 2.11
- Extension of the TwinCAT Simulation Manager user interface and many minor extensions.

2.1.2 New in Version 2.0

In the past the TwinCAT Simulation Manager ran the actual PLC/NC in a run-time system on the controller and the simulated process in another run-time system. Remote capability has now been integrated in Version 2.0 in order to ensure separation of the actual PLC/NC and the simulation process. The actual control system can thus run on the control computer and the simulation process on another PC. The link between control system and simulation process is realised in the so called "Distributed Simulation" via real-time Ethernet and the TwinCAT network variables - which guarantees a high performing deterministic data exchange.

Nevertheless if the Simulation runs on the same control computer, the realisation of the data transport is done transparently for the user via IO Process Image Mappings because these have less overhead in the local scenario. This is transparently hidden to the user and the appropriate data transfer protocol is chosen automatically by the TwinCAT Simulation manager dependant on the context - to reach the best possible performance.

For orchestration of the distributed control environment the TwinCAT Simulation Manager was extended with rich functionset and a lot of new user interface elements:

- A distributed navigation tree (Distributed configuration Navigator)
- Functions to assign and remove routes
- Browsing capabilities for the network (Intranet Broadcast search)
- Network debugging capabilities (Ping, ADS Ping), Controller State observer
- Caching and backup algorithms for (remote) Symbol und Controller configuration files
- Capabilities to assign network adapters for the simulation configuration
- Functions for Start, Stop and Restart distributed controllers orchestrated

Common Enhancements

- Not limited to one (local) simulation PLC anymore. Simulations can be freely assigned to controller PLC runtime systems.
- Support of .NET Framework 2.0
- Several enhancements of the User Interface

3 System requirements

- TwinCAT 2.10 Build 1325 or later
- Windows XP SP2, Windows Vista SP1 or later
- Microsoft .NET Framework Version 2.0 or later

4 Tutorials

4.1 Einführung

4.1.1 Profiles supported by the TwinCAT Simulation Manager

Two different configuration profiles are supported by the TwinCAT Simulation Manager (see Table 1).

Profile	Description
Original	All Devices and Hardware enabled (original configuration)
Remote	System is set to simulation mode. Not used or existant hardware is disconnected from the system and optionally replaced by simulation functionality.

Table 1: Profiles

The original profile is handled implicitly as the TwinCAT Systems configuration imported to the TwinCAT Simulation Manager. The Simulation profile is stored as a delta of Simulation actions referring to the Original profile.

The original <--> simulation mode switchover affects various components of the TwinCAT configuration:

- Replacement of IO by the simulation PLC (see Mappings of I/Os [▶ 14])
- Replacement of physically present axes by simulation axes (see Simulation axes [▶ 16])
- Deactivation of devices (see Deactivating devices [▶ 16])

Mappings of I/Os

Mappings of I/Os

The TwinCAT Simulation Manager follows the concept that non-available physical devices are substituted by one or more simulation PLCs. Device Signals that are transferred via IO-Hardware usually source or end up in PLC Symbols. In TwinCAT the binding between IO Hardware and PLC are realized by so called IO-Mappings.

Via its Simulation profile the TwinCAT Simulation Manager is capable to break up or probe the connections to this IO-Hardware and redirect them to a dedicated Simulation PLC. These simulation PLCs can run locally (Local-Simulation) or remotly (Remote-Simulation). The TwinCAT Simulation Manager internally chooses transparently from the user the appropriate data exchange protocol for symbol value mappings between the 'Original'- and 'Simulation-PLCs'. The data exchange occurs in realtime, so that the simulation is capable to emulate the machine characteristics (specifically the signal timing) to the highest degree possible.

Simulation Type	Description
Local	The Data-Transfer protocol is realized as TwinCAT internal IO-Mappings because they have the least overhead. The data exchange in this scenario is realized as IO-Driver synchronized Process-Image memory copy operations. The Realtime characteristics are directly derived from the Realtime characteristics of the TwinCAT Realtime system.
Remote	The data exchange is realized with EtherCAT Network variables (Master-Master communciation). Because the Ethernet-Network is involved, the overhead is higher. The Realtime characteristics are derived from the capabilites of the EtherCAT Master-Master communication.

Table 2: Simulation types

Various kinds of mappings are possible:

Type of mapping	Description
Observing	Observation of an I/O in simulation mode.

Type of mapping	Description
	A device input or output is observed. That is the device inputs or outputs are available at a simulation input and are available there for further processing. In original mode, the mapping to stimulation PLC does not exist.
Simulating	Simulation of an I/O, the mapping and the device are removed in simulation mode. The mapping between device input and the original PLC input (or device output and original PLC output) is separated and "simulated" by the simulation PLC. This means that for inputs the simulation PLC delivers the values for the original PLCs or for outputs the original PLC values are fed into the simulation PLC. In original mode, this "diversion" is inactive.
Original	No change in simulation mode. Communication between original PLC and device remains active (if device is not "deactivated", see Deactivating device [▶ 16]

Table 3: Types of data mappings

Original mode

Figure 1 [▶ 15] shows a TwinCAT system configuration in non-simulated (original) mode. The active mappings indicated with red and yellow run between the I/O devices and the original PLCs. The mappings indicated in grey to/from the simulation PLC are inactive.

This means that the original PLCs communicate with the I/O devices, which must be present and active. The simulation PLC is inactive and does not participate in the communication (original mode)

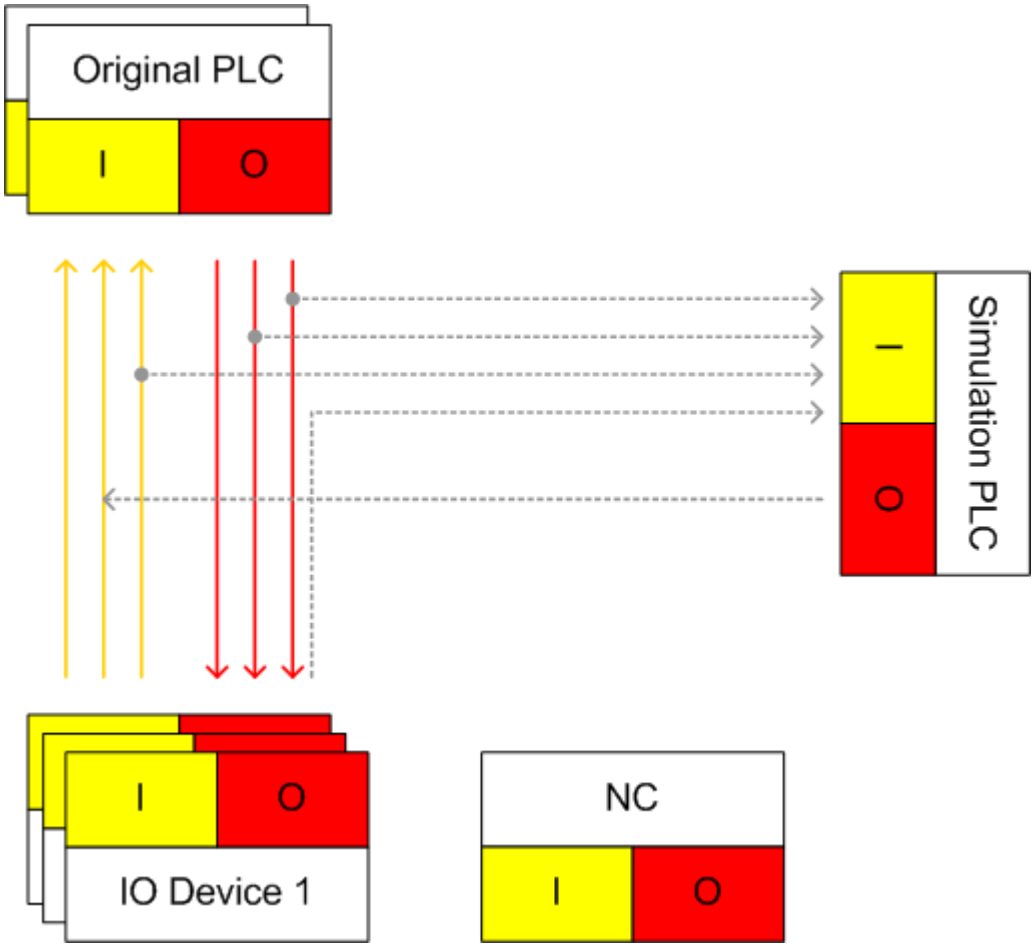


Fig. 2: Figure 1: TwinCAT System in original mode

The links / mappings from / to the Simulation PLC can represent local or remote connections, dependant of the location of the Simulation PLC.

Simulation mode

In simulation mode (see Figure 2 [▶ 16]), the simulation PLC is added. All mappings between original PLCs and I/O devices are distorted in such a way that devices to be simulated no longer participate in communication. All functions of the switched off devices that should be available in simulation mode, must be taken over by the simulation PLC.

The decoupled I/O devices are now no longer required for the (simulation) mode.

Simulated mappings break the mappings to the device and take over the original function. Observer mappings only feed process values into the simulation PLC (the original mapping is not broken).

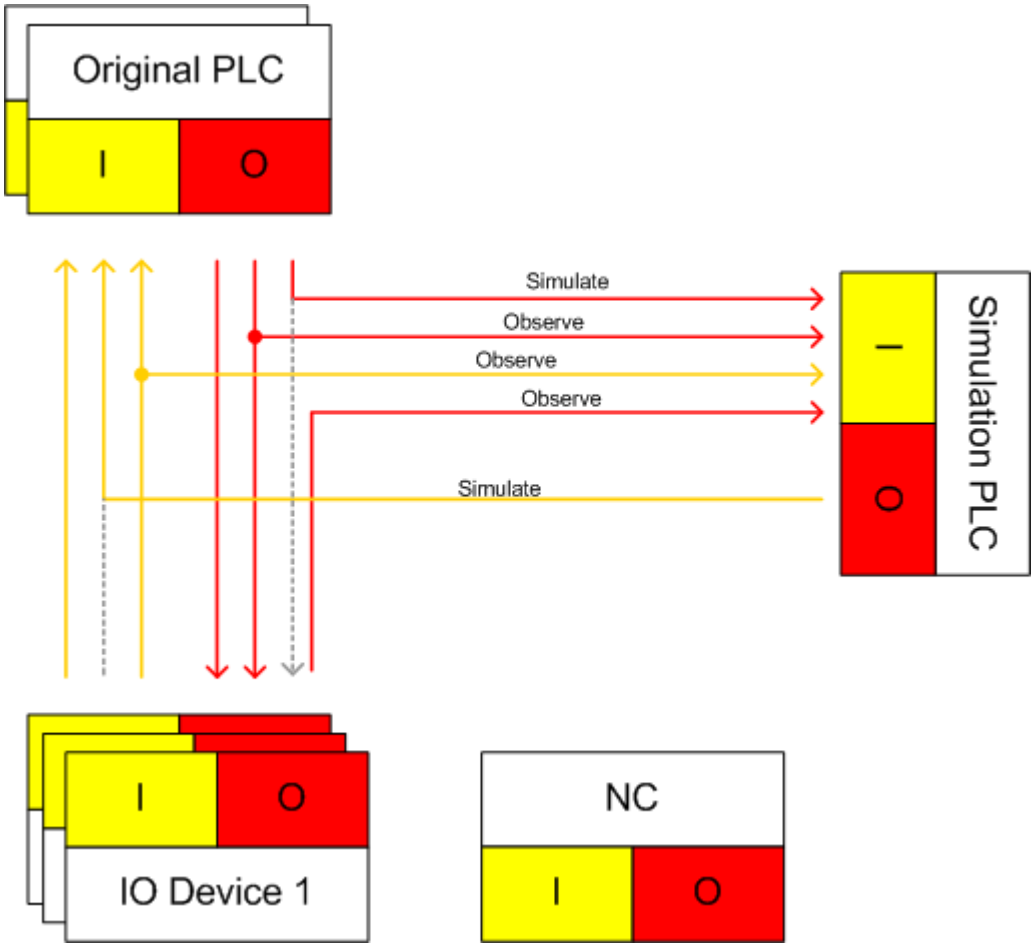


Fig. 3: Figure 2: TwinCAT system in simulation mode

The links / mappings from / to the Simulation PLC can represent local or remote connections, dependant of the location of the Simulation PLC.

Simulated (virtual) axes

Axes designated as 'simulated' are replaced in simulation mode. This means technically that the encoder of the axis will be replaced by a simulation encoder that is free running (setpoint equals currentPoint). The TwinCAT Simulation Manager persists the settings of the Original-Axis within the Original profile, so that all parameters can be restored when switching back to Original-Mode.

Deactivated devices

Devices designated as 'simulated' are deactivated in the simulation profile and don't necessarily to be physically present. The function of the device, insofar as required for the simulation, is taken over by the simulation PLC.

4.2 Projektvorbereitungen

4.2.1 Prepartions for creating a TwinCAT Simulation Manager project

Up from Version 2.1 of the **TwinCAT Simulation Manager** it is not necessary anymore that the involved TwinCAT devices run the necessary PLC programs and the necessary TwinCAT System Manager configuration before the project is created. System Manager configurations and PlcControl programs can now be imported and deployed to CPU Resources. Further information on this topic can be read in chapter Deployment [▶ 20].

The following hints for preparation of target systems, configurations and PLC-Programs should be understood as guideline. The **TwinCAT Simulation Manager** does not rely on these preparations, but they can simplify the workflow dependent of the use case.

PLC Programs and System Manager Configurations

All PLC Programs should reference the PLC Simulation Library "TcSystem.lib". Besides the essential base building blocks for simulation this library instantiates the status information for indicating the current simulation state within the PLC - simply by referencing the library.

Furthermore, it is helpful for the handling of the **TwinCAT Simulation Manager** tool to activate all included PLC projects as bootprojects (Figure 1 [▶ 17]). This enables a smooth switching process between 'Original' and 'Simulation'-Profiles. PLC programs will then restart automatically after every TwinCAT restart.

A selection of the "Remind to update boot project on change' feature within the PlcControl project options (Figure 2 [▶ 17]) will be helpful to remember that the bootprojects are staying consistent to the running PLC programs on the target.

Including the Symbol descriptions for the different PLC projects within the Simulation project is helpful. There exist two different procedures to make the symbol descriptions available in the **TwinCAT Simulation Manager project:**

1. If the *.pro and the corresponding *.tpy file is available on the development system (the system where the TwinCAT Simulation manager runs) then these files are copied to the project.
2. The symbol information was copied to the target system during Project download from the PLC control. To enable this feature the Option "Download Symbol description" must be activated in the PlcControl options (Figure 2 [▶ 17]) and the compiled project must be available on the target system.

To generate a Simulation PLC Framework project automatically from the original PLC template it is necessary to make the symbolic information available with procedure 1 or 2. If the Simulation PLC program will be coded manually, no access to the symbolic information is necessary.

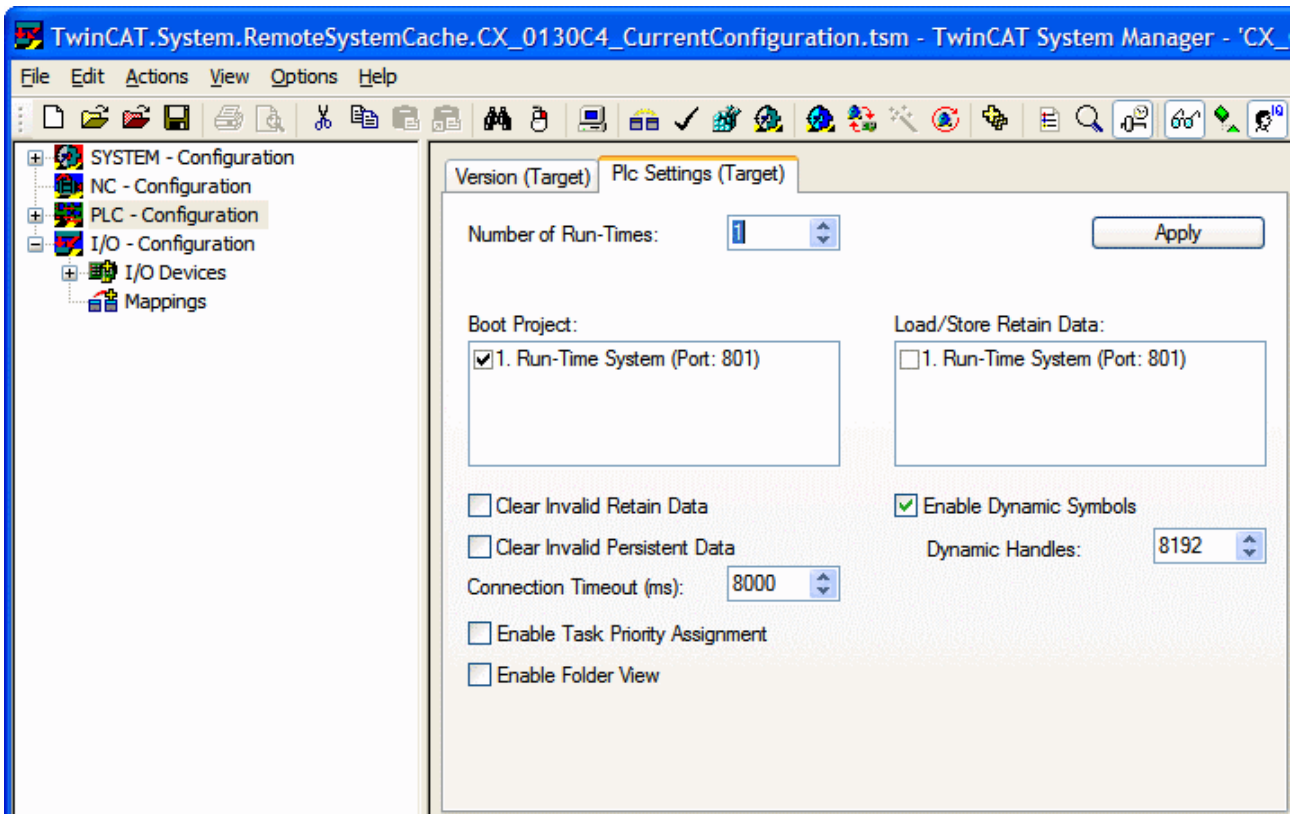


Fig. 4: Figure 1: Bootproject activation for PLCs (TwinCAT System Manager configuration).

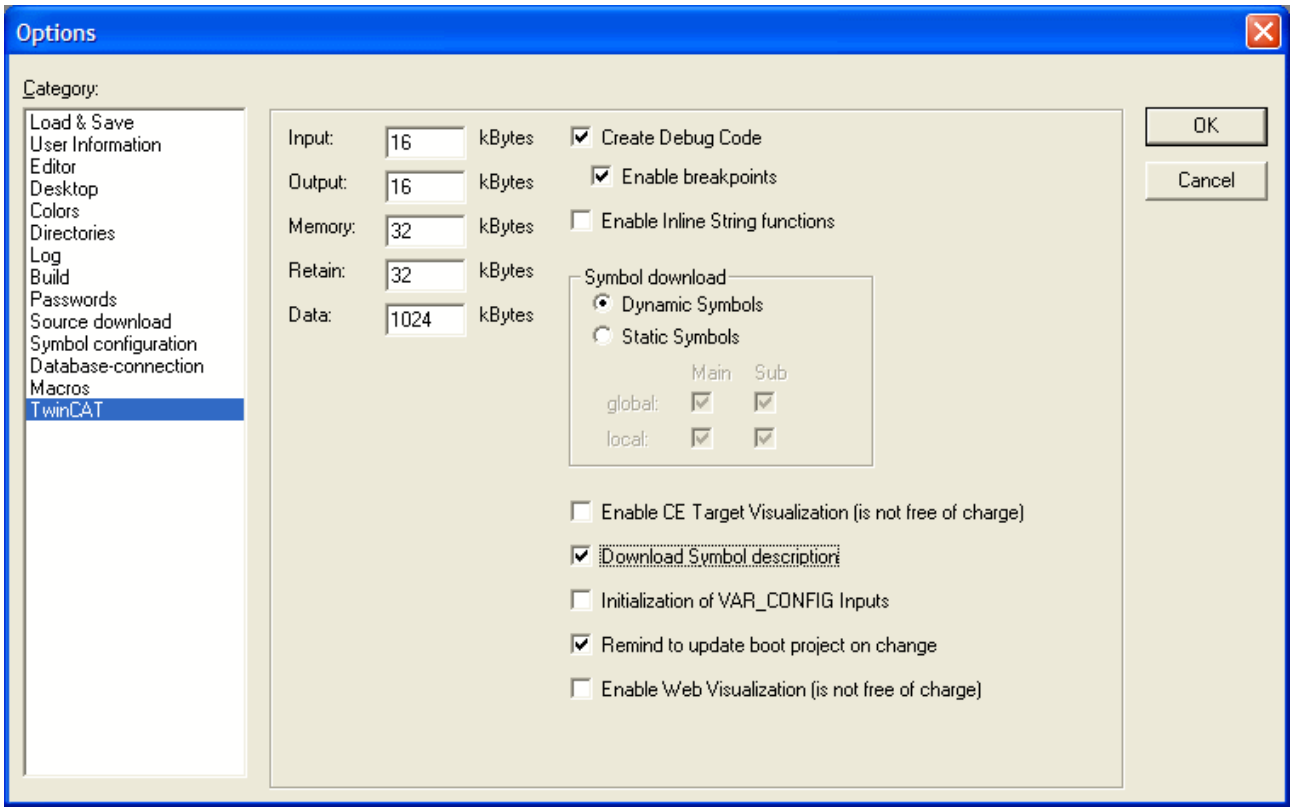


Fig. 5: Figure 2: Activation of the Symbol description download and the update boot project reminder (TwinCAT PlcControl options)

Target Systems

The TwinCAT System Manager offers the option to store the activated configuration on the target system (Option "Auto Save to target" Figure 3 [▶ 19]). This option can be helpful because it actualizes the configuration during every activation on the target. If the configuration is available on the target, there is no need to import the configuration to the simulation project.

During selection of the target system within the "Add System ..." dialog (Figure 4 [▶ 19]) of the **TwinCAT Simulation Manager** there is the option to upload the target-site stored configuration (Checkbox "Load configuration from target system"). The (nonrecurring) upload takes place if the Dialog is confirmed and a copy of the configuration will be created within the simulation project.

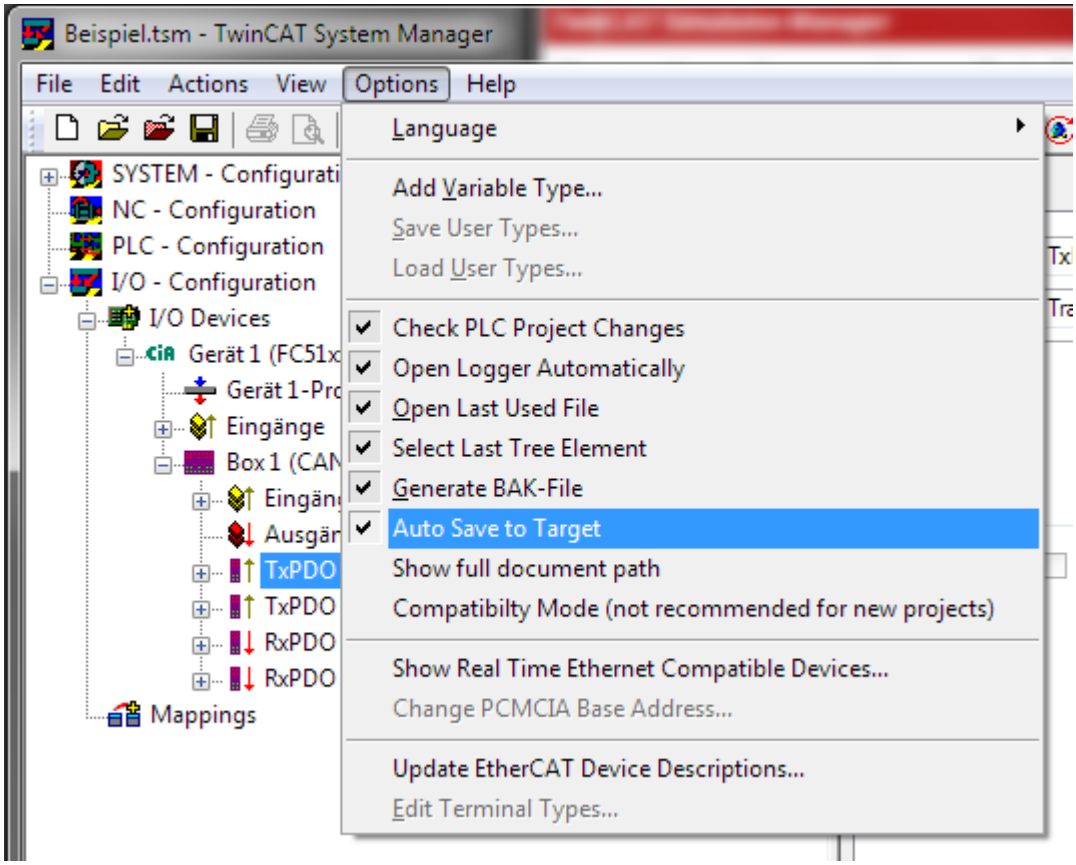


Fig. 6: Figure 3: Automatic storing of the current configuration on the target system (System manager option).

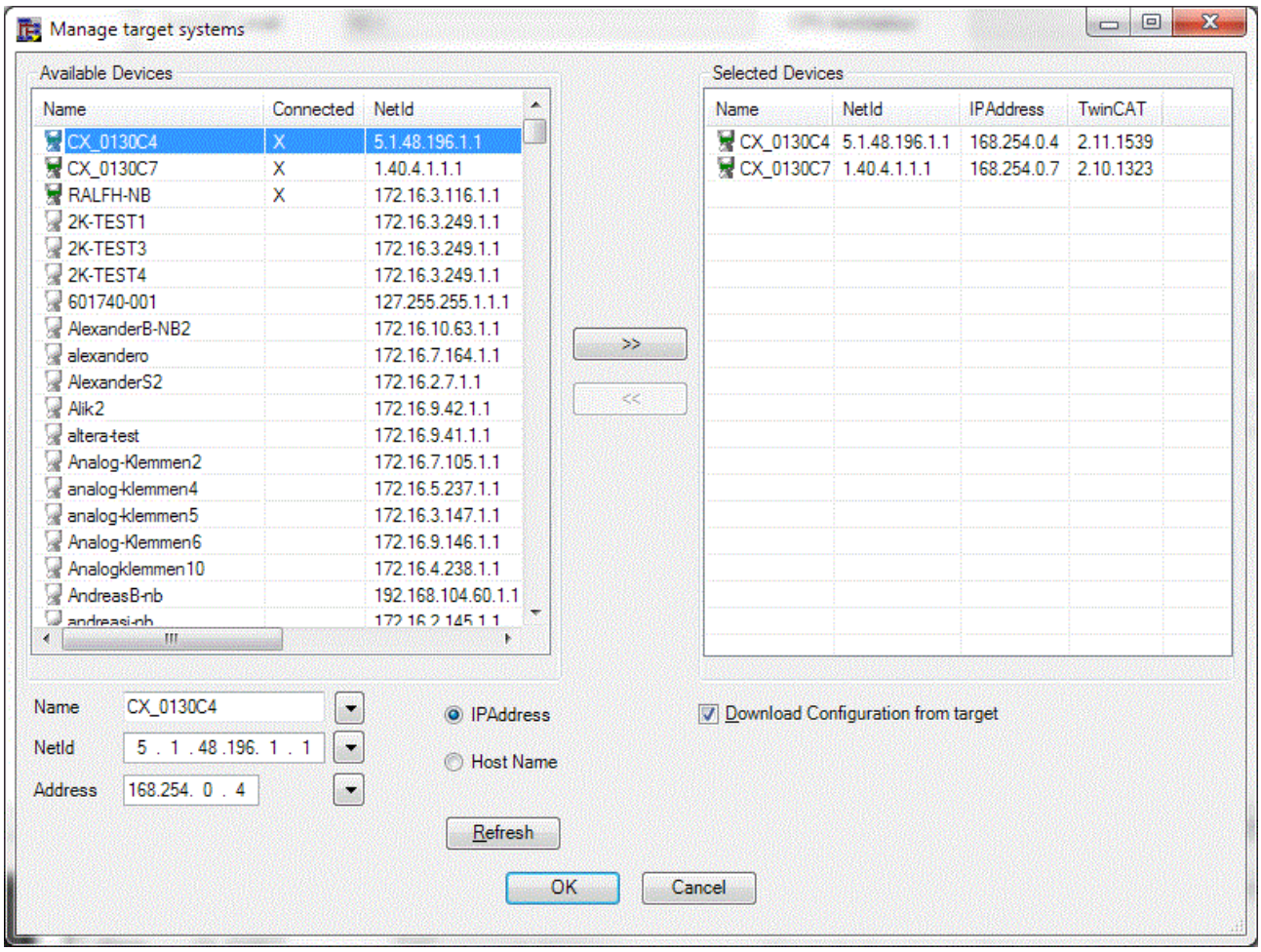


Fig. 7: Figure 4: Selection of the CheckBox 'Download Configuration from target' to download target site stored System Manager configurations (Simulation Manager Dialog)

4.3 Ressourcenverteilung und Zielsysteme festlegen

4.3.1 Deployment and selection of the Target Systems / Devices

Concept

The **TwinCAT Simulation Manager** supports local as well as distributed scenarios. To deploy realtime functionalities in an environment form one to multiple resources (CPU), the application realizes the following groups of features:

- Selection of target systems [▶ 21]
- Assignment of configurations [▶ 23]
- Assignment and deployment of PLC-Programs [▶ 24]

The system can scales from pure local to full distributed environments. To maintain all relevant data of a simulation-project at a central location (mostly on the configurator PC), all the project data will be stored within the Simulation Manager project directory. This folder will then act as central data store. All referenced System Manager configurations and PLC Programs will be copied into the project cache during creation of the Simulation Manager project. The Application then acts on these copies, the originals are not touched anymore.

Selection of target systems

The Dialog 'Manage target systems' [▶ 54] is opened via main menu 'Edit -> Select target Systems ...' (Figure 1 [▶ 39]).

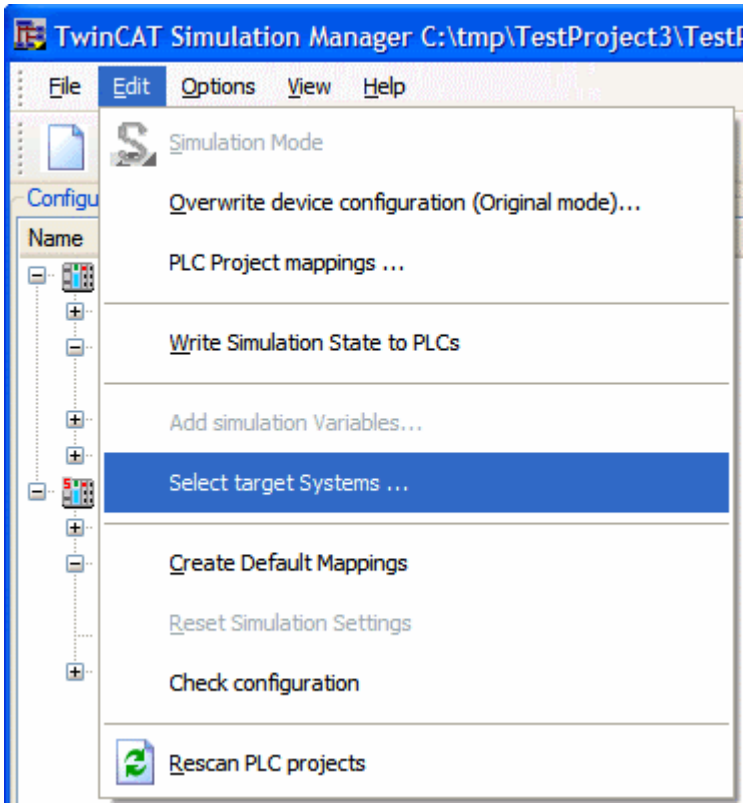


Fig. 8: Figure 1: Calling the "Select target systems" dialog.

The Dialog consists mainly of two list boxes (see Figure 2 [▶ 54]):

- On the left side, all found target (TwinCAT) systems within the connected Ethernet network are listed (Broadcast search)
- On the right side all participating systems of the TwinCAT Simulation Manager configuration are listed.

The list of targets within the TwinCAT Simulation Manager Configuration can be changed via the Buttons ">>" and "<<".

With pressing the "OK" Button, the list is accepted and the TwinCAT Simulation Manager loads all Current Configurations from the target systems plus the available PLC Symbol files and stores them in it's local cache for access. Afterward the Navigation window [▶ 39] is updated.

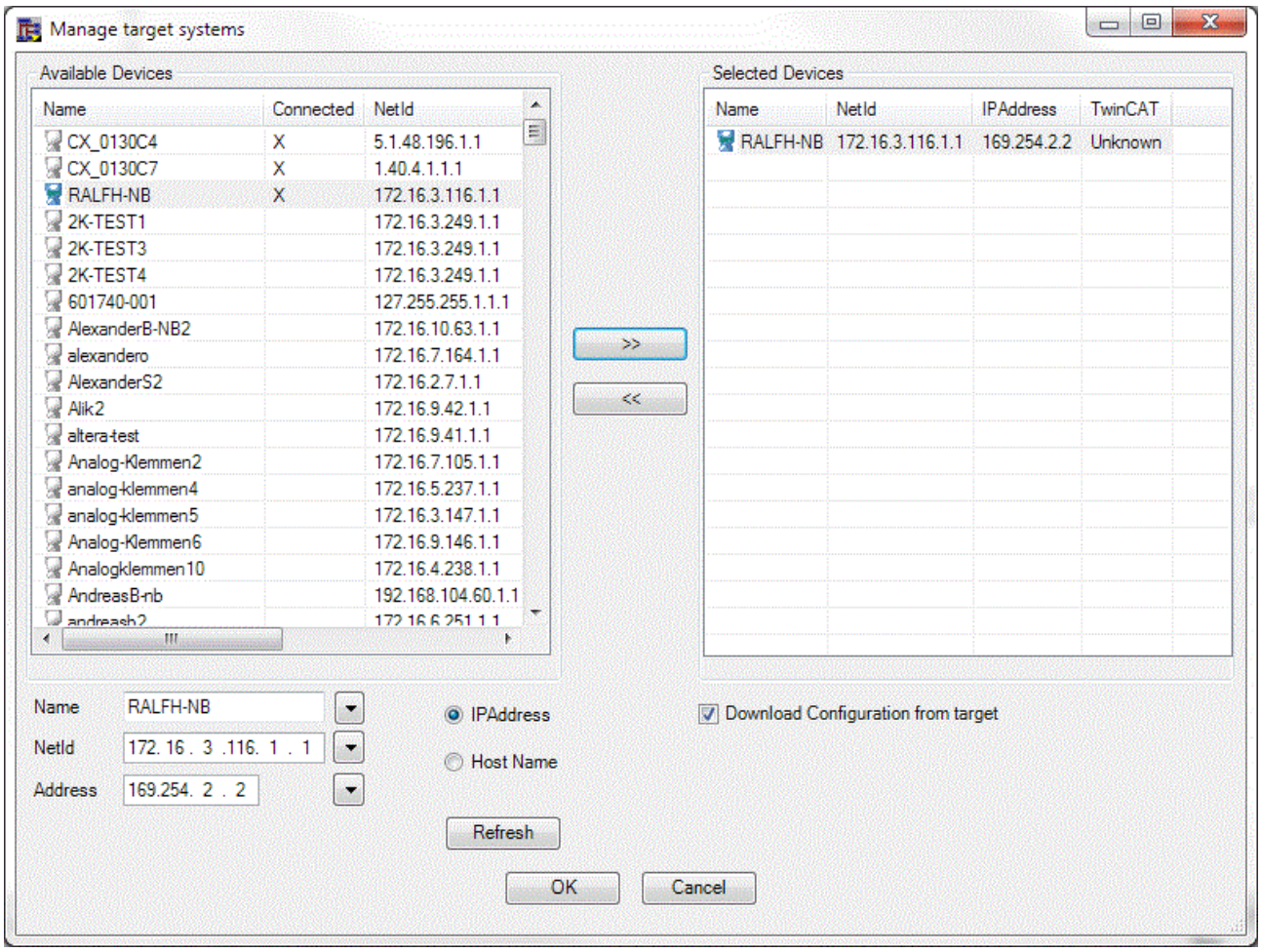


Fig. 9: Figure 2: Selection of the target systems

Assigment of Configurations

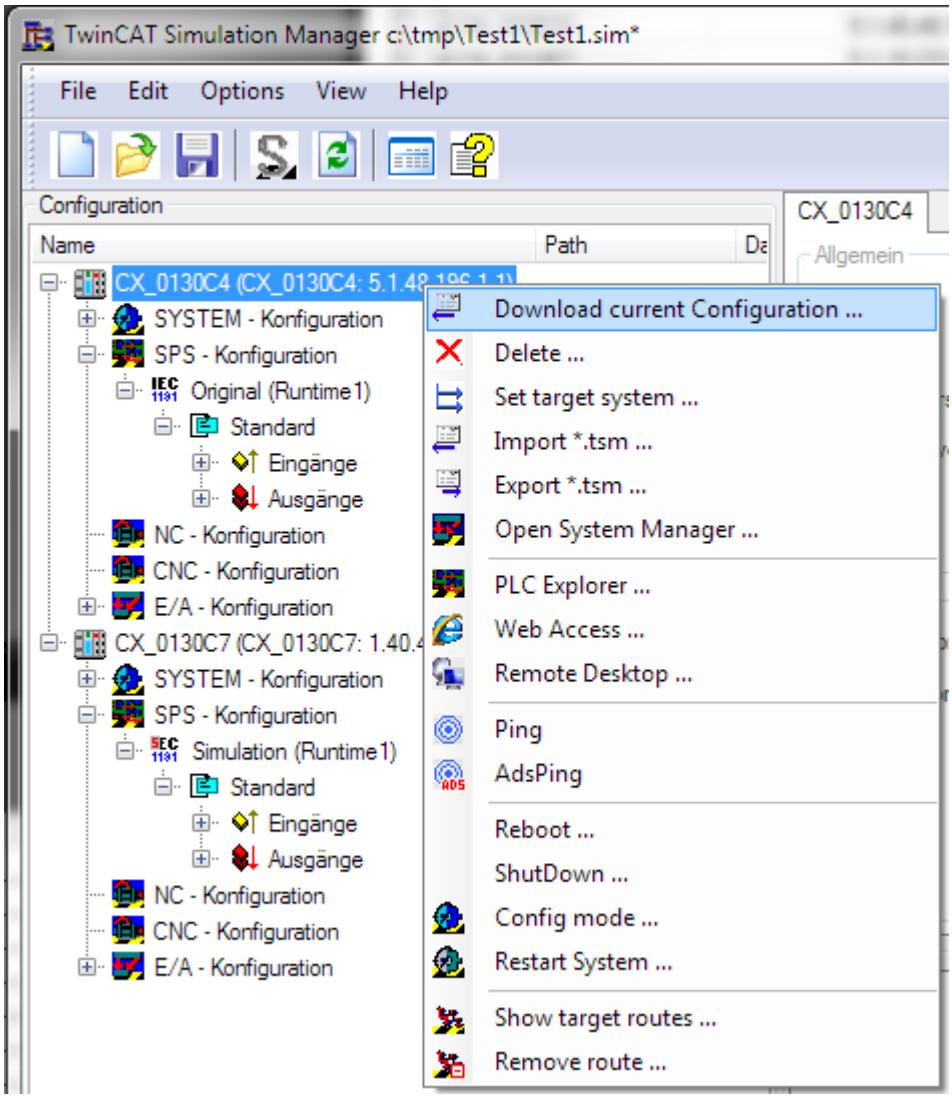


Fig. 10: Figure 3: Context-menu entries for System Manager Configuration assignment (target system related)

Entry	Description
Download current Configuration ...	Downloads the current configuration from the selected target system and updates/overwrites the simulation-projekt cached data.
Delete ...	Removes the target system and its configuration from the simulation project.
Set target system ...	Sets a new address at the selected target configuration. The configuration will be assigned to a new target device.
Import *.tsm ...	Imports a configuration into the simulation project and sets the target system address of this imported configuration to the selected target address.
Export *.tsm...	Copies the selected System Manager configuration from the simulation project to an external folder.

Table 2: Context-menu entries for System Manager Configuration Assignment (target system related)

Assignment and deployment of PLC-Programs

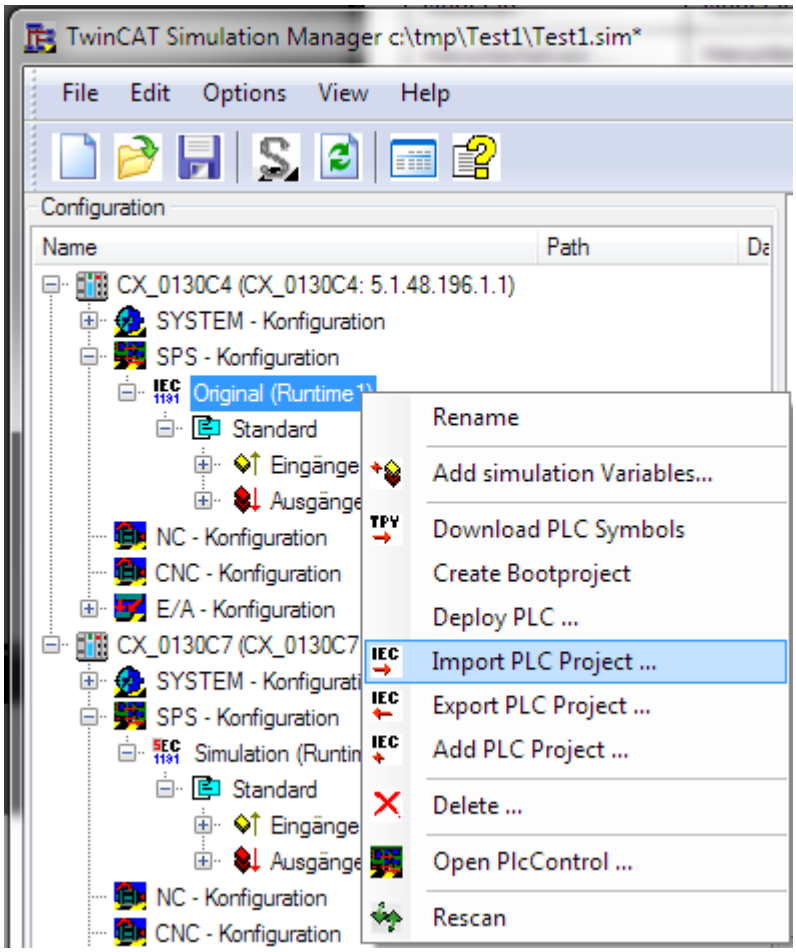


Fig. 11: Figure 4: Context-menu entries for PLC-Program assignment and deployment

Entry	Description
Create Bootproject	Creates the PLC program as bootproject on the target device.
Deploy PLC ...	Shifts the selected PLC-Program to another available runtime.
Import PLC Project ...	Imports a PLC-Program to the selected PLC-Runtime. The existing PLC-Program will be overwritten.
Export PLC Project ...	Export the PLC-Program from the selected PLC-Runtime into an external folder.
Add PLC Project ...	Adds a new PLC-Program to an available/free PLC-Runtime.
Delete ...	Removes the specified PLC-Program from the simulation configuration.

Table 3: Context-menu entries for PLC-Program assignment and deployment

4.4 Assignment of Simulation-PLCs

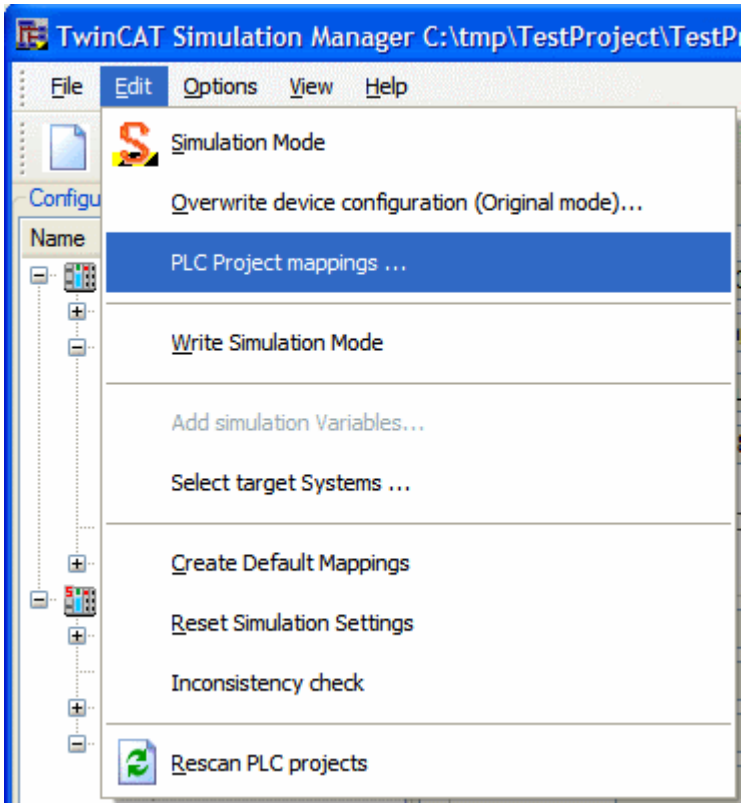


Fig. 12: Figure 1: Calling the PLC Assignment Dialog from main menue

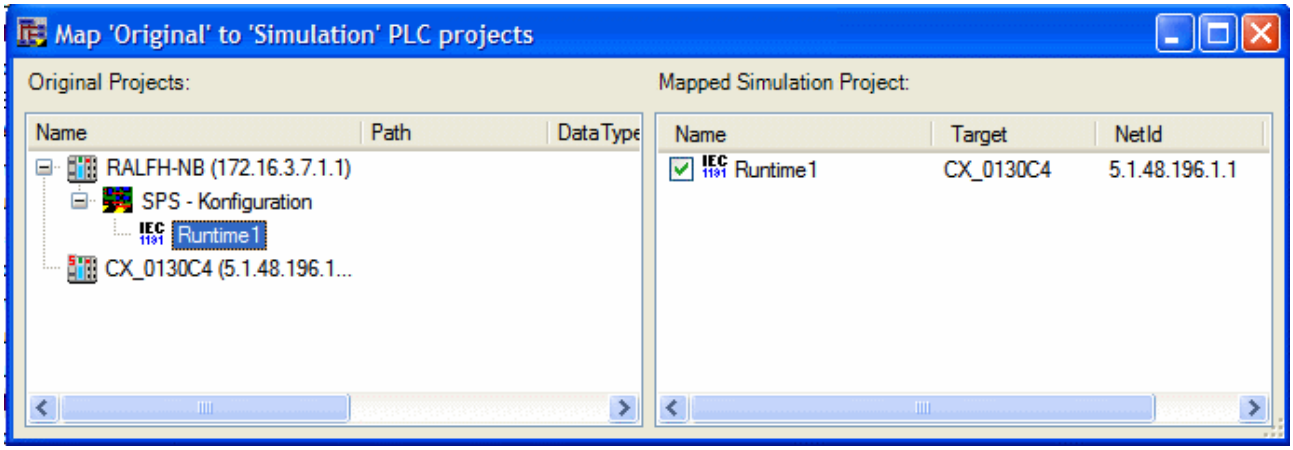


Fig. 13: Figure 2: Assignment of Original-PLC to Simulation-PLC

4.5 Activation of the simulation mode

When the simulation button is clicked in original mode, the following dialog wizard (Figure 1) is opened.

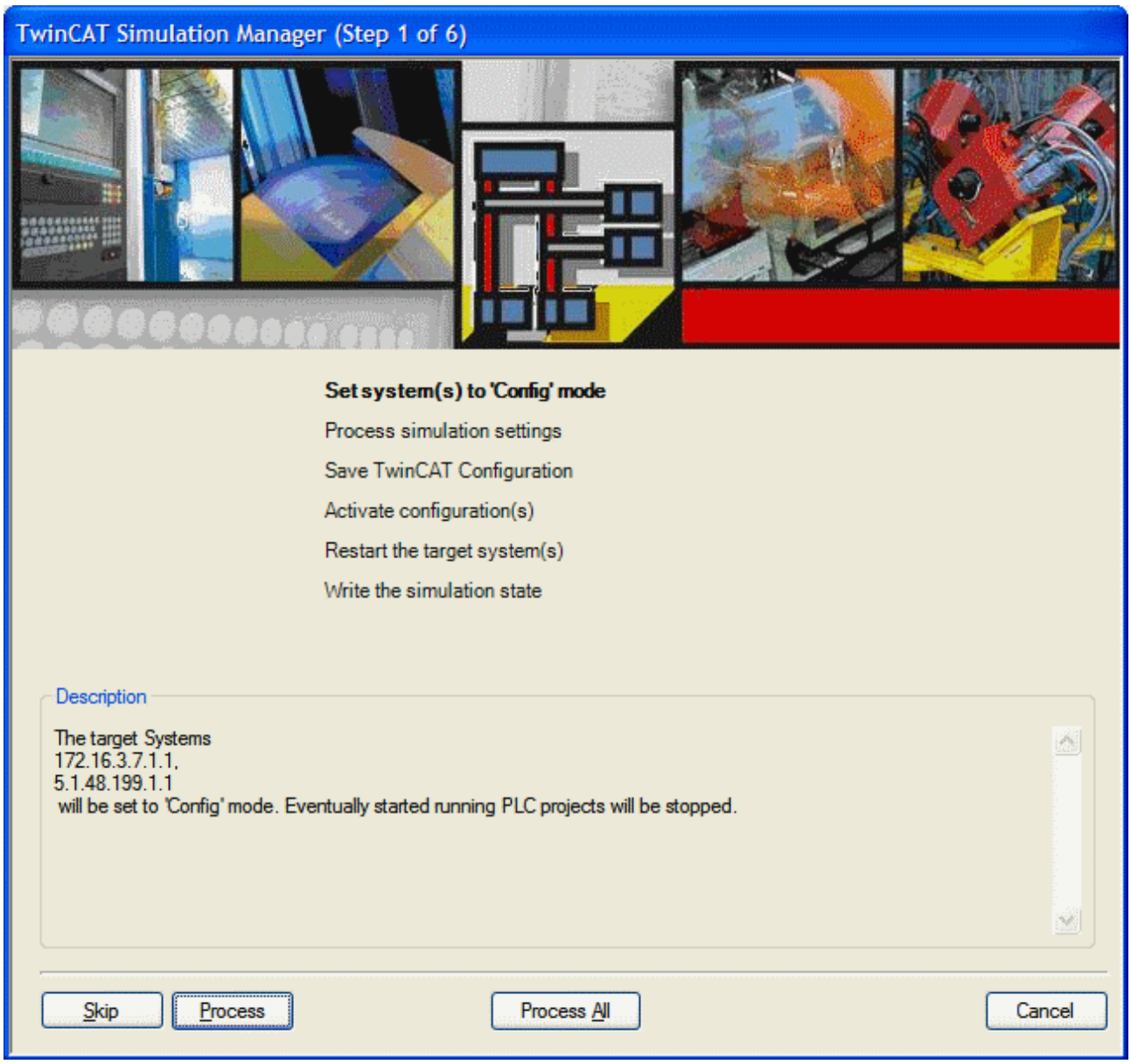


Fig. 14: Figure 1: Simulation On Wizard User Interface

The description of the command buttons of the Wizard is given in Table 1.

Push-Button Name	Description
Skip	Skips the step (only if it is optional)
Process	Processes one single step and stops afterwards
Process All	Processes all steps. If any step fails, then the processing will be stopped
Cancel	Cancels the further processing of the steps and closes the Wizard-Dialog
Finish	All steps processed successfully and the "Finish" button closes the Wizard-Dialog
Redo	Appears when one step fails. This gives the opportunity to repeat the actual step.

Table 1: Buttons on the Wizard

The switchover to simulation mode is performed in several steps (Figure 1 and Table 2). Some of these steps are mandatory and some are optional (that is they can be bypassed with the 'skip' button).

Requirements

Step	Name	Description
1	Set system(s) to 'Config' mode	All participating systems will be set to 'Config' mode. If the local system is one of these systems, the TwinCAT Simulation manager first sends the 'Reconfig' command to this system. If the local system reaches the 'Config' mode, the 'Restart' command is send to all other remote systems at the same time. The Step succeeds if all systems are in 'Config' mode.
2	Process simulation settings	The current I/O mappings are linked (locally or remotely) to the simulation PLC, axes are switched over to simulation mode and (simulated) devices are deactivated! The created simulation profile is not used if it is not activated and until the TwinCAT system has performed a restart. The step is succeeded when no error occurred setting the configurations.
3	Save TwinCAT Configuration(s)	Saves the locally cached TwinCAT System Manager Configuration files of all participating systems.
4	Activate Configurations(s)	Activate the configurations on the targets. The configurations will remain inactive until a TwinCAT restart is carried out on the target. The command will succeed if the activating processes ran through without errors.
5	Restart TwinCAT System(s)	Restarts the TwinCAT target systems with the activate configurations. Like in Step 1, first the (optional) local system will be restarted and as second step all 'remote' ones in parallel. This step is succeeded if all targets are in 'Run' mode.
6	Write the simulation state	Resets the 'dwSimState' Global PLC Variable to '1' for all participating PLCs. This Symbol is the indicator for the PLCs that the system is in "Original" or "Simulation" mode. The step succeeds if the targets are reachable, the Symbol (TcSimManager.lib) is contained in the PLC project and the value could be written with success.

Table 2: Process Steps to change from Original Mode --> Simulation Mode

If all steps succeeded the wizard looks like shown in Figure 2.

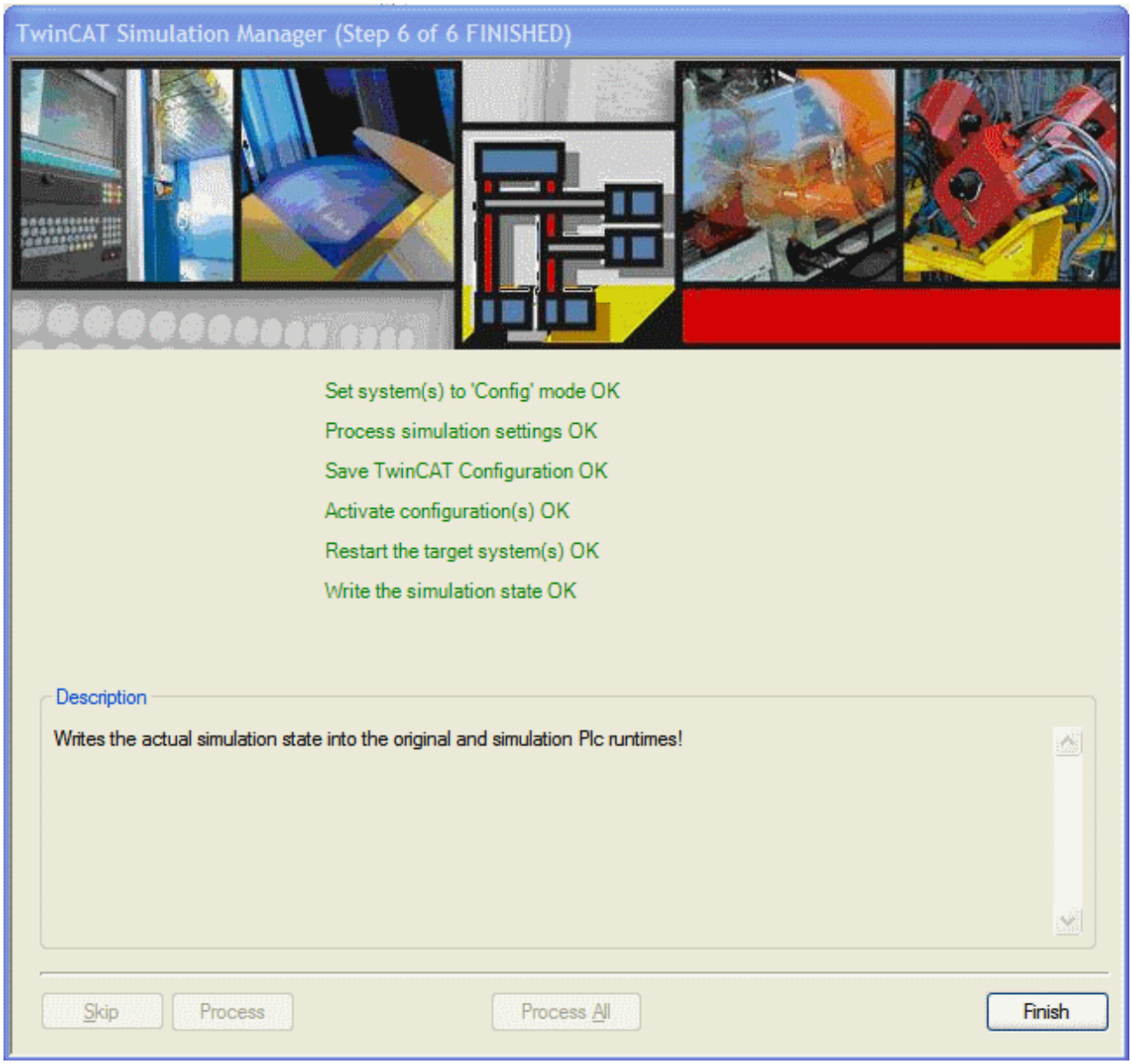


Fig. 15: Figure 2: Wizard Dialog when successfully set to the simulation mode

The wizard can now be closed via the "Finish" button.

4.6 Deactivation of the simulation mode

When the simulation button is clicked in simulation mode, the following wizard dialog (Figure 1) is opened.

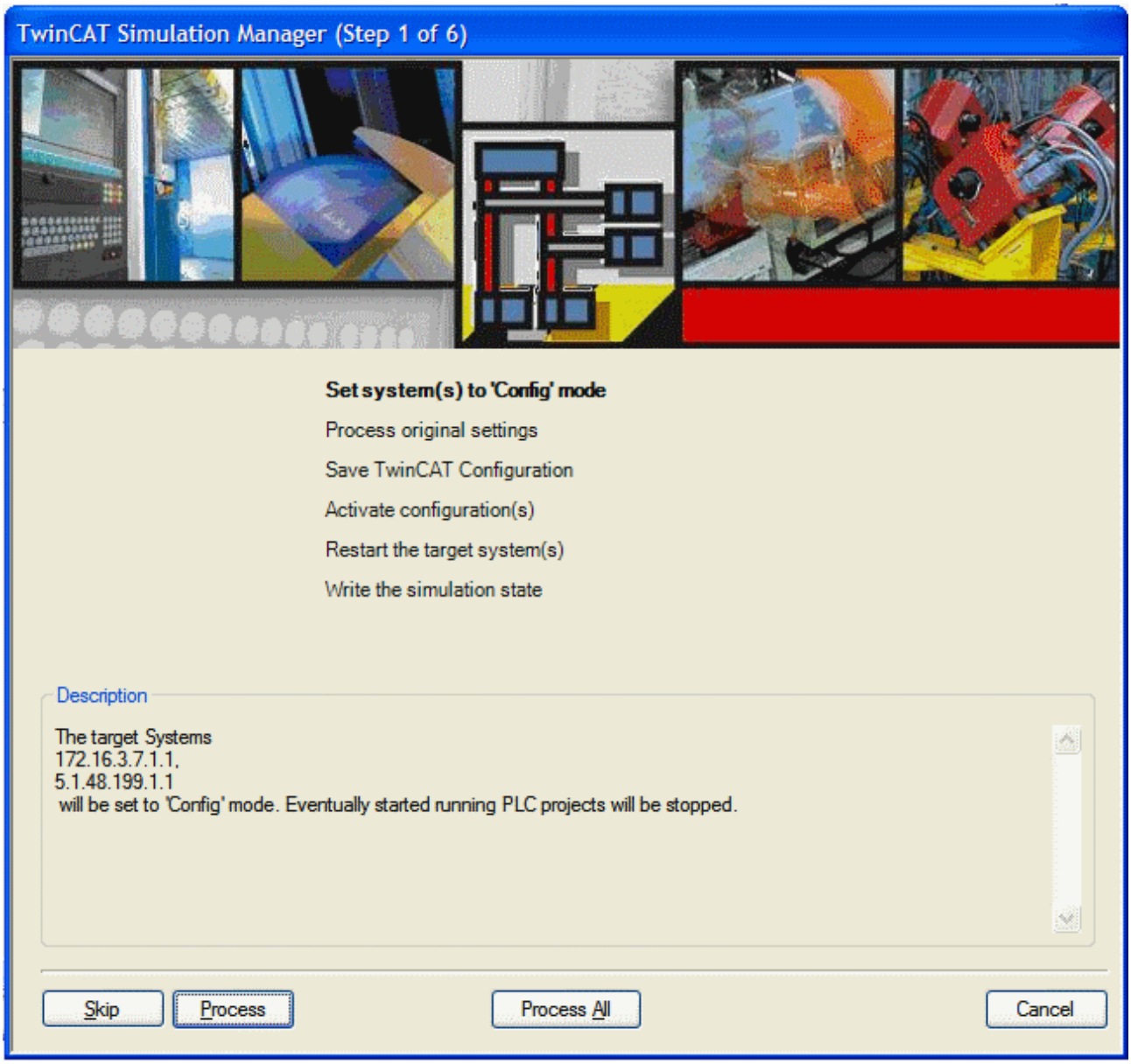


Fig. 16: Figure 1: Simulation Off Wizard User Interface

The description of the command buttons of the Wizard is given in Table 1.

Push-Button Name	Description
Skip	Skips the step (only if it is optional)
Process	Processes one single step and stops afterwards
Process All	Processes all steps. If any step fails, then the processing will be stopped
Cancel	Cancels the further processing of the steps and closes the Wizard-Dialog
Finish	All steps processed successfully and the "Finish" button closes the Wizard-Dialog
Redo	Appears when one step fails. This gives the opportunity to repeat the actual step.

Table 1: Buttons on the Wizard

The switchover to original mode is performed in several steps (see Figure 1 and Table 2). Some of these steps are mandatory and some are optional (that is they can be bypassed with the 'skip' button).

Requirements

Step	Name	Description
1	Set System(s) to 'Config' mode	All participating systems will be set to 'Config' mode. If the local system is one of these systems, the TwinCAT Simulation manager first sends the 'Reconfig' command to this system. If the local system reaches the 'Config' mode, the 'Restart' command is send to all other remote systems at the same time. The Step succeeds if all systems are in 'Config' mode.
2	Process original settings	The current I/O mappings are removed (locally or remotely) from the simulation PLC, axes are switched back to original mode and deactivated devices enabled again The created original profile is not used if it is not activated and until the TwinCAT system has performed a restart. The step is succeeded when no error occurred setting the configurations.
3	Save TwinCAT Configuration(s)	Saves the locallly cached TwinCAT System Manager Configuration files of all participating systems.
4	Activate Configuration(s)	Activate the configurations on the targets. The configurations will remain inactive until a TwinCAT restart is carried out on the target. The command will succeed if the activating processes ran through without errors.
5	Restart the target system(s)	Restarts the TwinCAT target systems with the activate configurations. Like in Step 1, first the (optional) local system will be restarted and as second step all 'remote' ones in parallel. This step is succeeded if all targets are in 'Run' mode.
6	Write the simulation state	Resets the 'dwSimState' Global PLC Variable to '0' for all participating PLCs. This Symbol is the indicator for the PLCs that the system is in "Original" or "Simulation" mode. The step succeeds if the targets are reachable, the Symbol (TcSimManager.lib) is contained in the PLC project and the value could be written with success.

Table 2: Process Steps to change mode from Simulation Mode --> Original Mode

If all steps succeeded the wizard looks like shown in Figure 2.

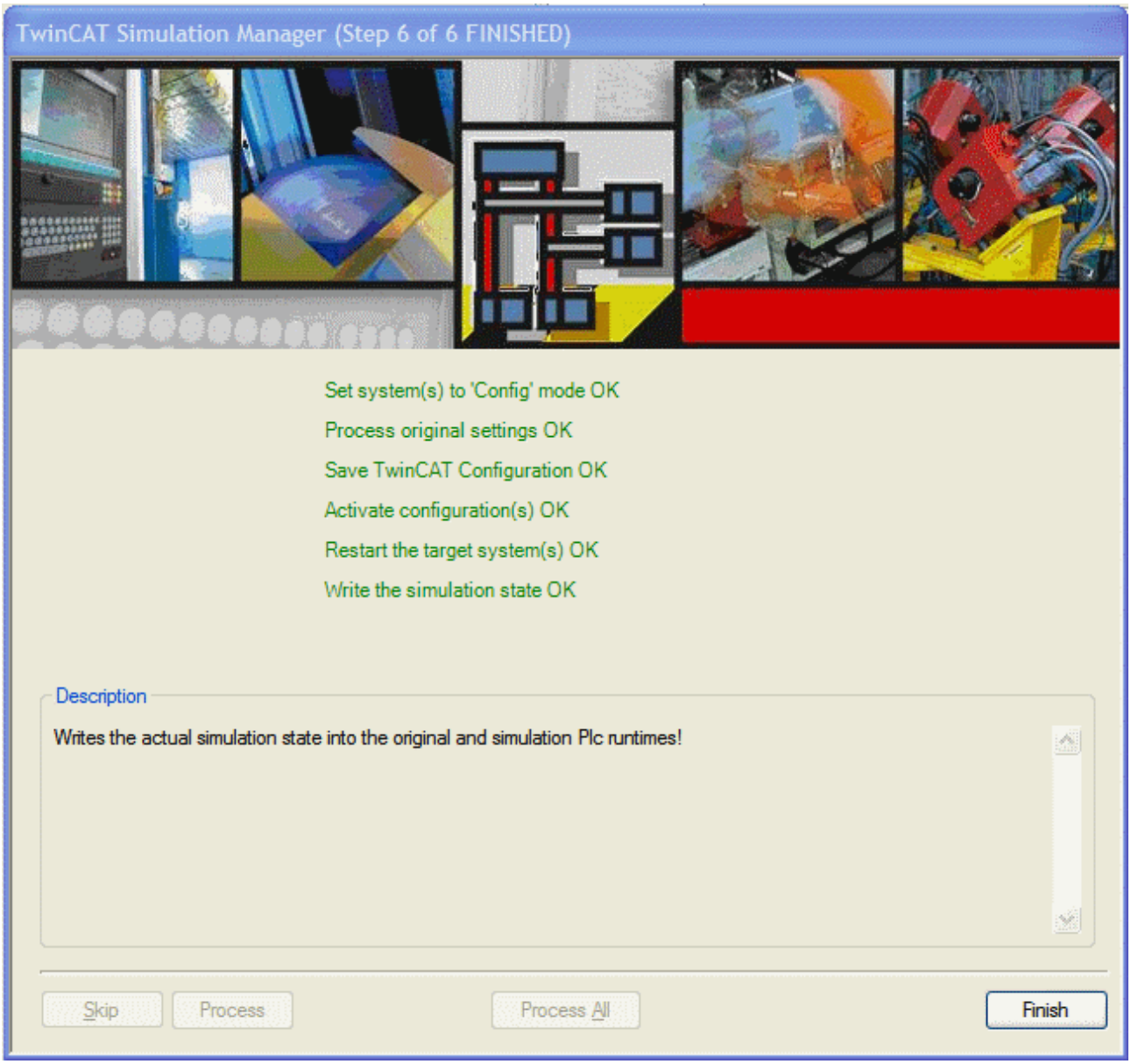


Fig. 17: Figure 2: Wizard Dialog when successfully returned to the original mode

The wizard can now be closed via the "Finish" button.

4.7 Creating simulation template project

Creating and adding a new simulation PLC is done via selection and activating the Contextmenu entry "Create Simulation PLC-Project" of the PLC runtime that is intended to be simulated in the configuration navigation tree (see Figure 1).

Within the now appearing Dialog, the Simulation PLC runtime must be selected (Figure 2). This dialog shows the currently available target systems. If the designated simulation system is not in the tree, it can be added via the "Search" button. With leaving the dialog over "OK" the TwinCAT Simulation Manager generates a simulation framework for the selected PLC, assigns it to the specified target system and runtime and adds it to the TwinCAT Simulation Manager configuration project.

This PLC framework project will contain all located IO-Symbols of the original PLC project in reversed/ mirrored allocation (inputs --> outputs and outputs --> inputs, see Table 1).

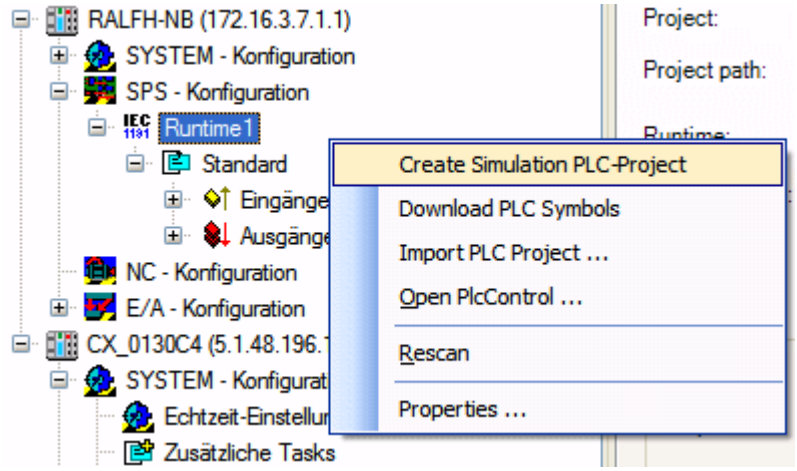


Fig. 18: Figure 1: Creation of a Simulation PLC Project via Context menu

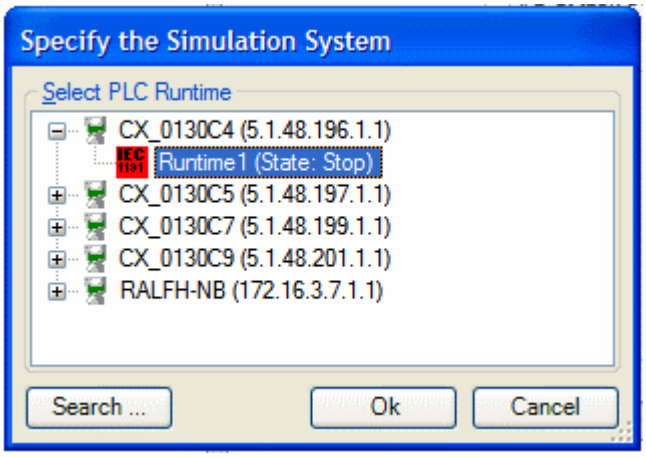


Fig. 19: Figure 2: Selection of the PLC target sytem

A template PLC project can be created via the menu item 'Edit' --> 'Generate simulation PLC project'. This template is created from the currently integrated original PLC projects. The following components of the original project(s) are considered and possibly adopted in the simulation PLC project.

- Libraries that are in the standard PLC control library path *("[INSTALLATIONSPFAD_PATH]\Plc\Lib"]*
- All allocated input and output symbols (%I, %Q) of the original PLC projects are adopted in the simulation project. Inputs however are created as outputs and according to the rules defined in the table. Permanently allocated, universal variables and block variables as well as VAR_CONFIGs are used as symbols (also see 'Standard mappings [▶ 35]').

	Original PLC symbol	Device I/O	Signal direc-tion	Simulation PLC symbol	Description
1	VAR_INPUT Allocation (%I)	Input	<--	VAR_OUTPUT (%O)	A device delivers a signal to the PLC, e.g. an "On" switch. In the simulation mode, the simulation PLC delivers the switch position via an output symbol to the original PLC.
2	VAR_OUTPUT Allocation %Q	Output	-->	VAR_INPUT (%I)	A block output signal is transmitted to a device, e.g. a 'bRunning' flag to a light
3	VAR_INPUT Allocation %Q	Output	-->	VAR_OUTPUT (%I)	The symbol specifies the interface to a device input. The block is not usually filled with code

	Original PLC symbol	Device I/O	Signal direc-tion	Simulation PLC symbol	Description
4	VAR_OUTPUT Allocation %I	Input	<--	VAR_INPUT (% Q)	The symbol specifies the interface to a device output. The block is not usually filled with code

Table 1: Generation rules for "reverse" or "mirrored" symbolic

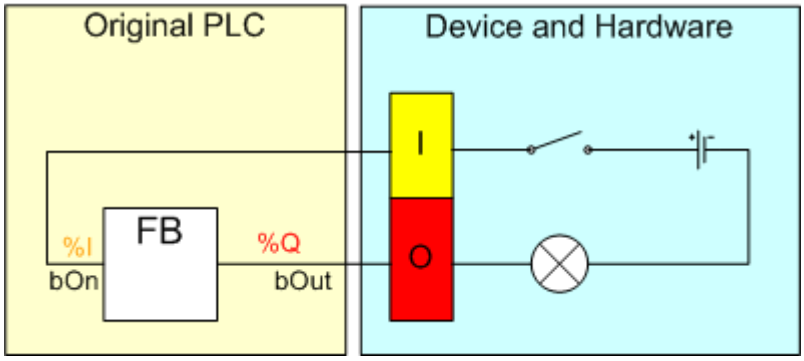


Fig. 20: Figure 3: Case 1 and 2 (original mode)

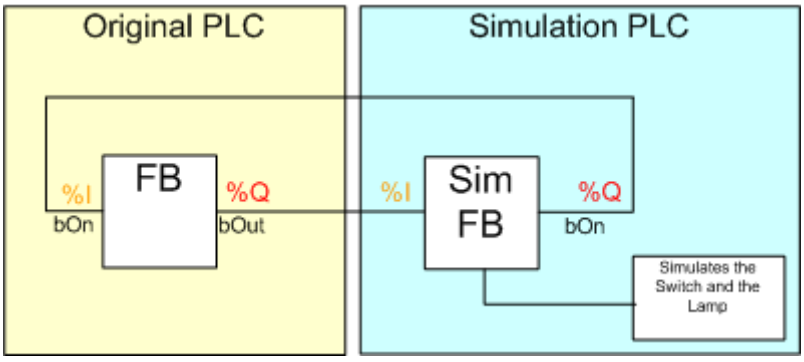


Figure 4: Case 1 and 2 (simulated)

- The data types of the allocated instances / symbols
- POUs, which contain allocated symbols.
  POU blocks may contain other POUs. These are recursively created when they contain allocated symbols. In the process, the inputs and outputs and their allocations are mirrored.
  The creation only takes into account symbols / instance paths that lead to a allocated symbol. Other variables will not be created. Symbols that are added to the orignal project after the Simulation configuration has been created, can be maintained into the simulation project via the Adding symbols to the simulation project [▶ 34] functionality.

Restrictions:

- One original PLC project can be bound to exactly one Simulation PLC project.
- Located Symbols out of Libraries are not supported
- Located Pointer data types are not supported

The next step is to fill the empty shell of the simulation project with simulation code. This is specific to the application. For this task PLC Control Application can be opened via context menu "Open PlcControl ...".

In the simulation mode, the original symbols with their correspondences are connected in the simulation PLC.

4.8 Simulationssymbole hinzufügen

4.8.1 Adding symbols to the simulation project

The "Add Symbols to Simulation" dialog offers the possibility to maintain the simulation project after changing the original project. Before opening the dialog the TwinCAT Simulation manager scans the original and the simulation PLC project and offers the delta of the symbols inside the left hand side list box ("Original PLC(s)").

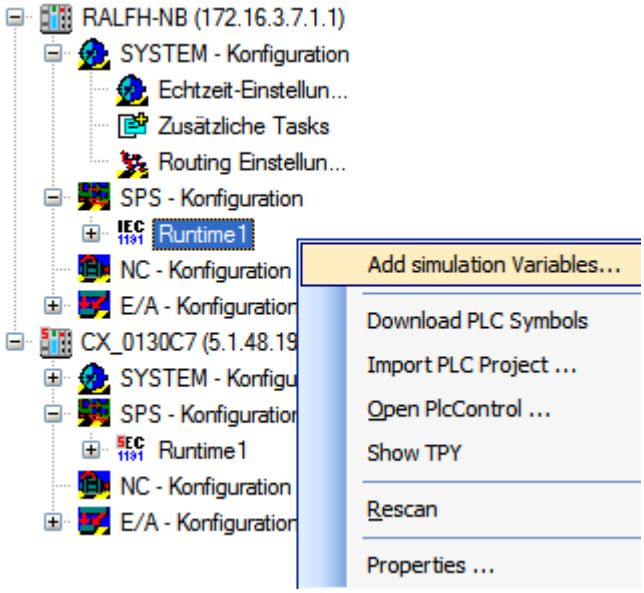


Fig. 21: Figure 1: "Add simulation Variables ..." Context menu on the PLC Runtime node

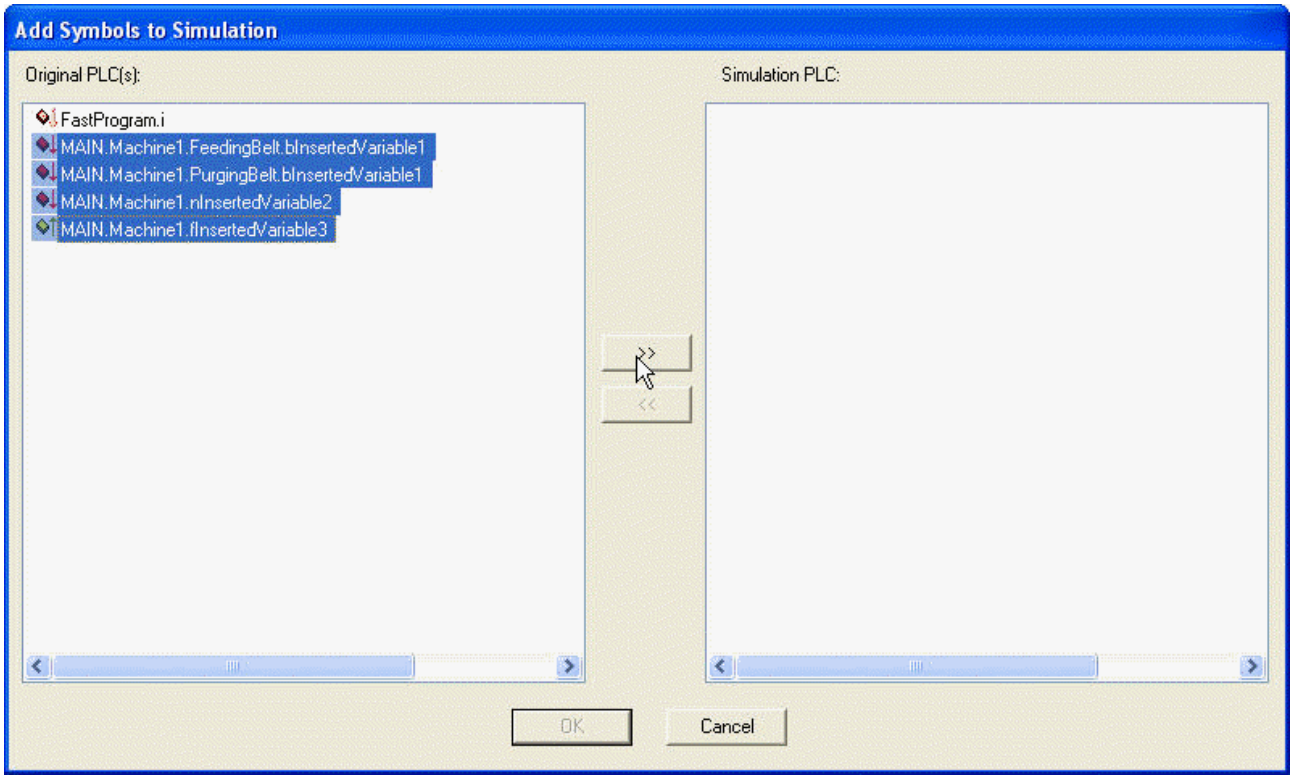


Fig. 22: Figure 2: Selection of the Symbols to add within the simulation project

With adding the symbols to the right hand side pane ("Simulation PLC") the symbols to add can be selected.

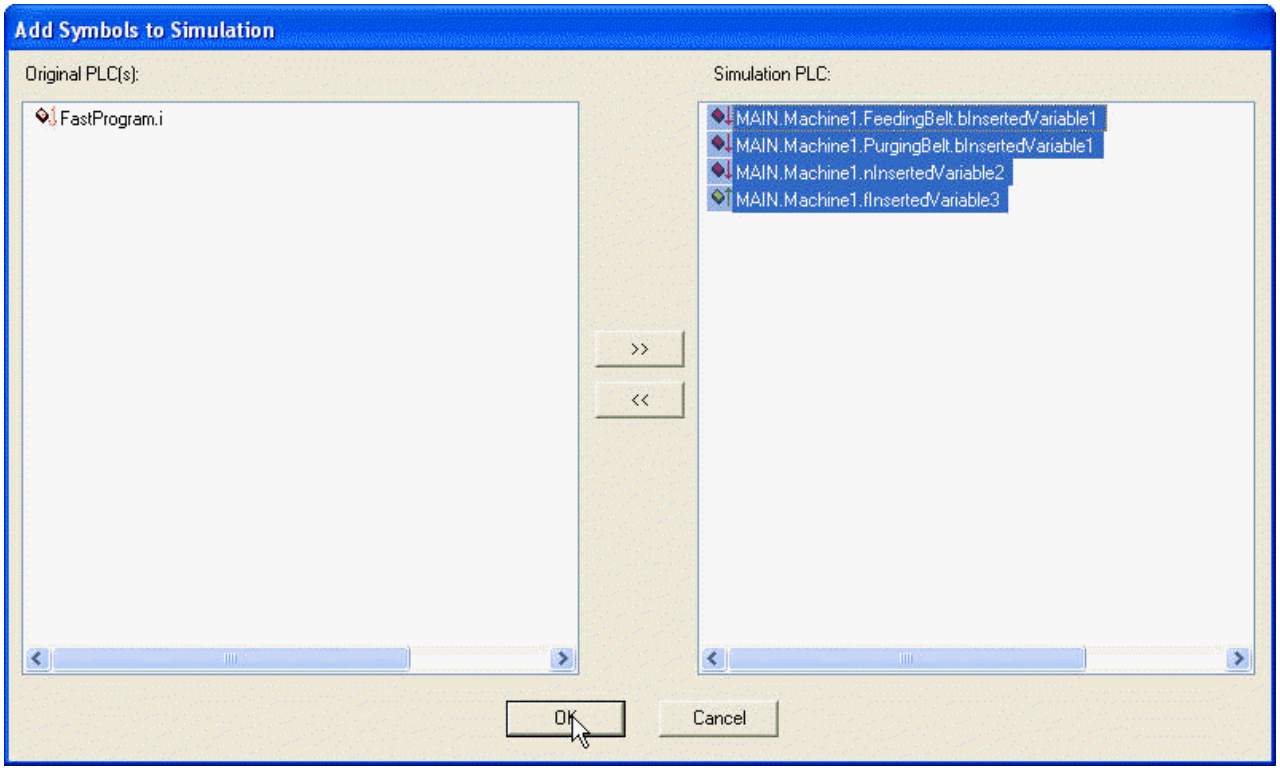


Figure 3: Movement of the selected Symbols to the Simulation PLC ListView

Selecting "OK" accepts the dialog setting, opens the PLC Control compiles the Simulation project. After the "PLC Control" program is closed again, the Simulation Manager should reflect the changes in its Navigation window [▶ 39].

4.9 Default (standard) mappings

Standard mappings are mappings that can be automatically set up by the simulation manager in one step. An automatic mapping offered for symbols that are available both in the original PLC and in the simulation PLC (with the same data type) is created by calling the main menu item 'Create Default Mappings' from the 'Edit' menu (see Figure 1).

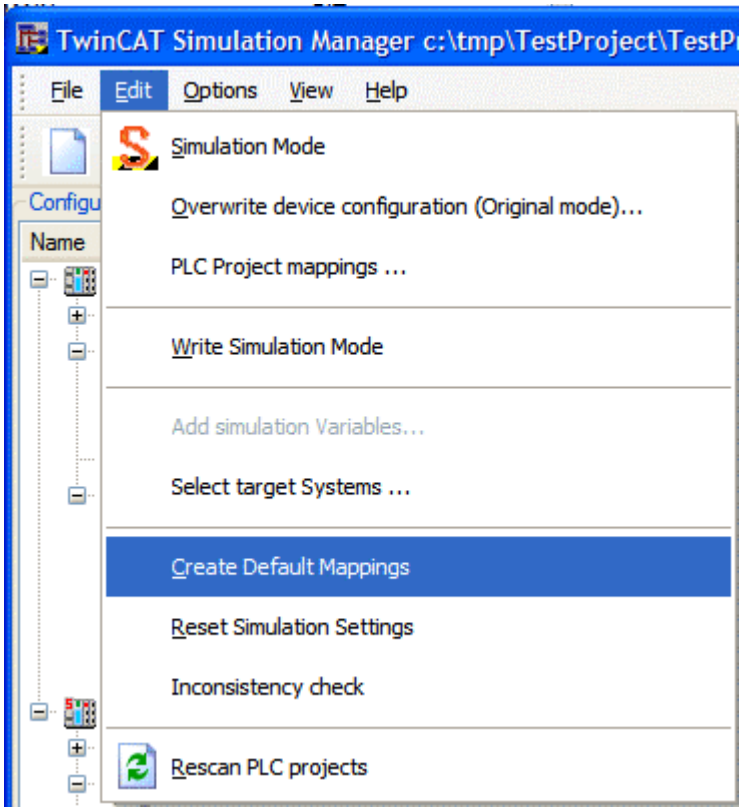


Fig. 23: Figure 1: Calling "Create default mappings"

This functionality iterates over all Symbols within the original PLC projects and tries to bind it to simulation symbols of the same symbolic name / access path, same data type but reversed location (reversed %I / %Q). If an applicable simulation symbol exist, an local (IO-Mapping) or remote Mapping (via Realtime Network Variables) will be created within the simulation settings.

Standard mappings are very useful when used in conjunction with the Create Simulation project [▶ 31] feature. This feature ensures that the symbolic names and access paths within the simulation project are easily connectable to the original PLC project for the user and automatic program support by ensuring identical naming conventions.

In Figure 2 the navigator window has the original Symbol 'bRun' selected. The 'Simulation Targets View' on the right hand side shows the possible connections within the assigned simulation PLC project. The Default Mapping for this symbol is 'MAIN.bRun' and is marked with green text.

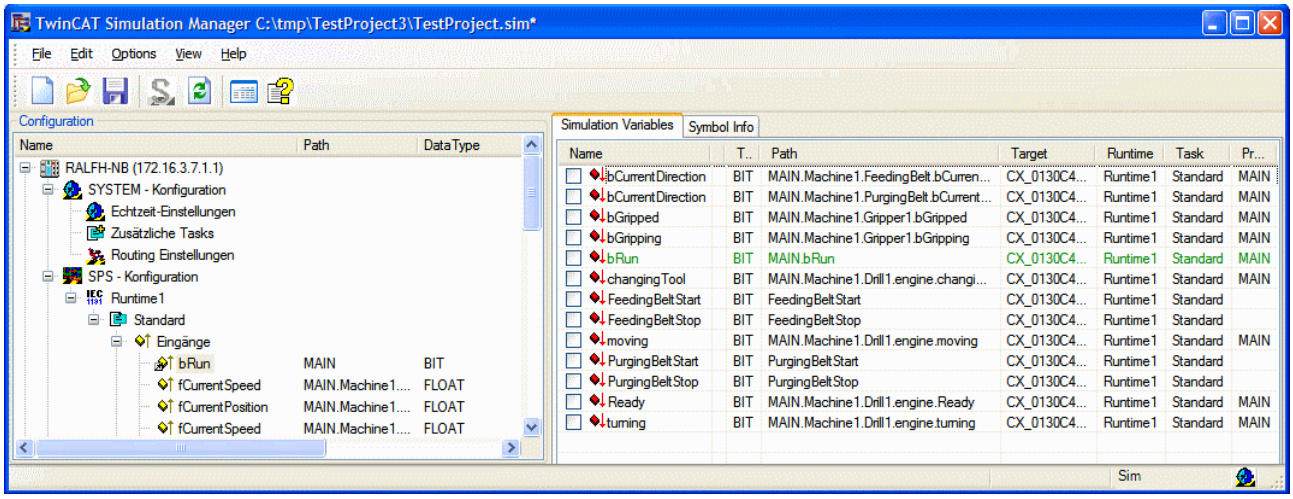


Fig. 24: Figure 2: Simulation Mapping Targets View: Before creating the Default Mappings

If the default connections are generated, the original symbol MAIN.bRun will be bound to the simulation symbol 'MAIN.bRun' for the simulation profile (see Figure 3). Simulated Symbols will be marked with the red 'S' within its icon to show that a simulation setting is assigned.

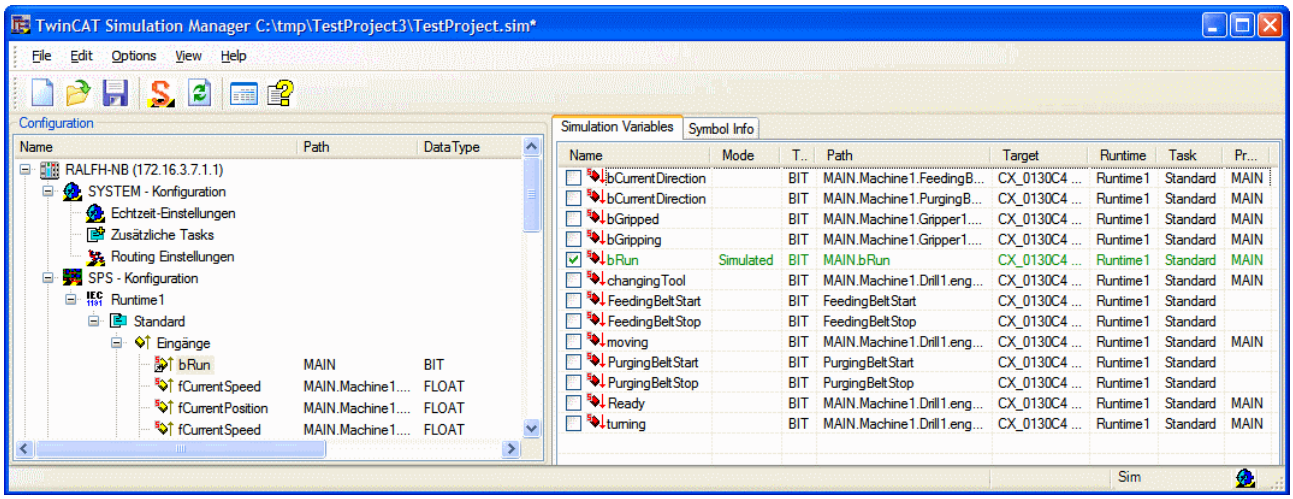


Fig. 25: Figure 3: Simulation mapping Targets View: After creating the Default mappings

5 User interface

5.1 Views

5.1.1 Main screen

The main screen is divided into three separate areas:

- The navigation tree [▶ 39] (similar to the system manager configuration tree, under the heading "Configuration").
  This view is to navigation through the TwinCAT Simulation Manager configuration ordered by the included target systems.
- Simulation configuration window (context dependent, in Figure 1 the Symbol Target Window [▶ 45] is shown), to set the simulation properties
- Current Mappings View [▶ 47] ("Mappings") for showing the currently set mappings of the selected symbol.

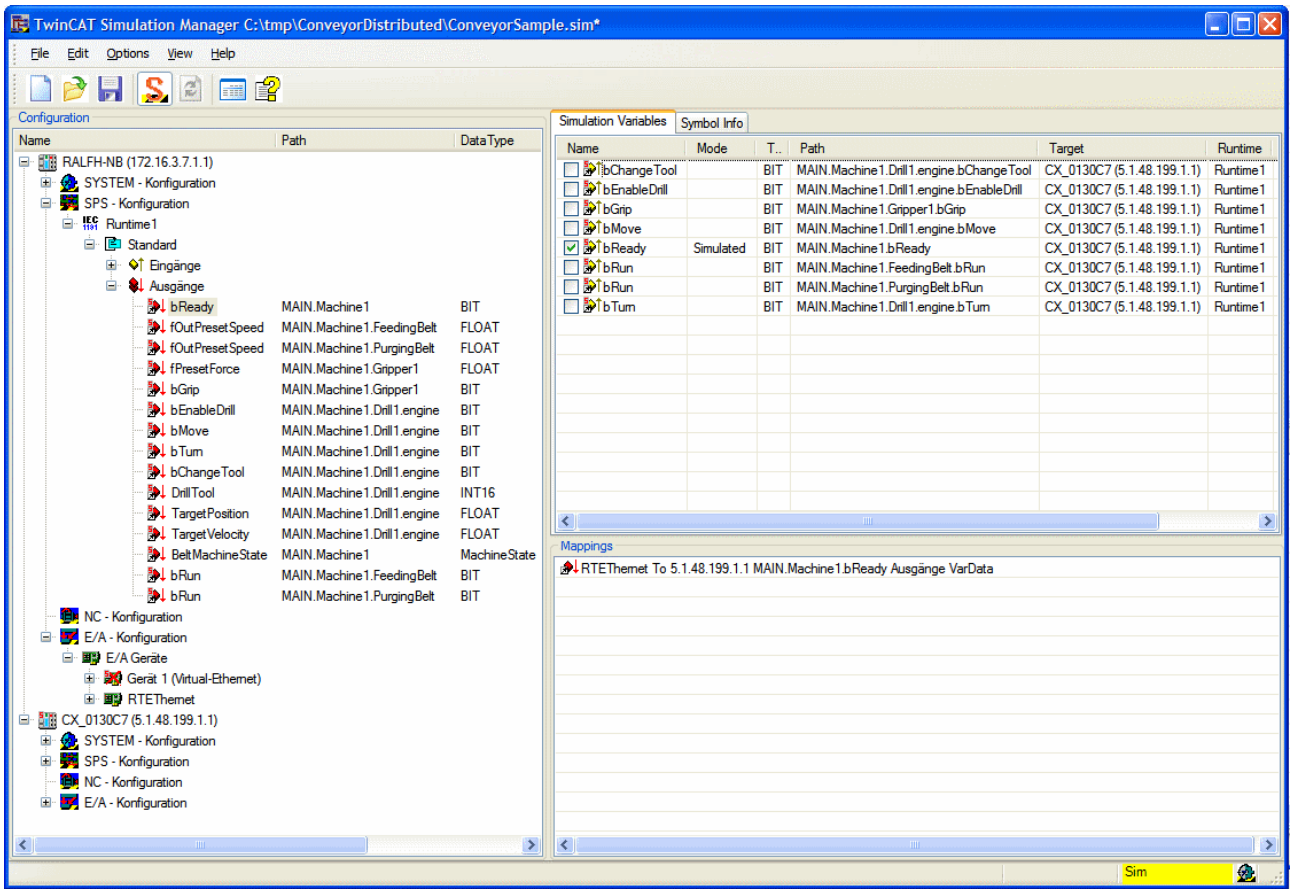


Fig. 26: Figure 1: TwinCAT Simulation Manager screen

When the simulation mode is switched on, this is indicated by yellow flashing "Sim" in the status bar of the simulation manager.

The TwinCAT Icon at the right border of the status shows the mode and availability of the local TwinCAT System.

5.1.2 Navigation Window

The 'Navigation Window' is the centralized control interface of the TwinCAT Simulation Manager. It embraces all integrated functionalities and enables the browsing through the TwinCAT Simulation Manager configuration. The shown navigation tree is closely harmonized with the configuration tree of the TwinCAT System Manager so that the user recognizes the structure of the machine configuration - but is extended in terms of supporting distributed systems and simulation configurations.

The principle tasks of the 'Navigation Window' are:

- Navigating through the distributed system and accessing further programming / configuration and debugging tools and applications
- Management of the Original and Simulation profile
- Configuring and generating simulations for various components (e.g. axes, PLCs, IOs and Devices)

An example for a navigation tree shown in the 'Navigation window' can be seen within Figure 1.

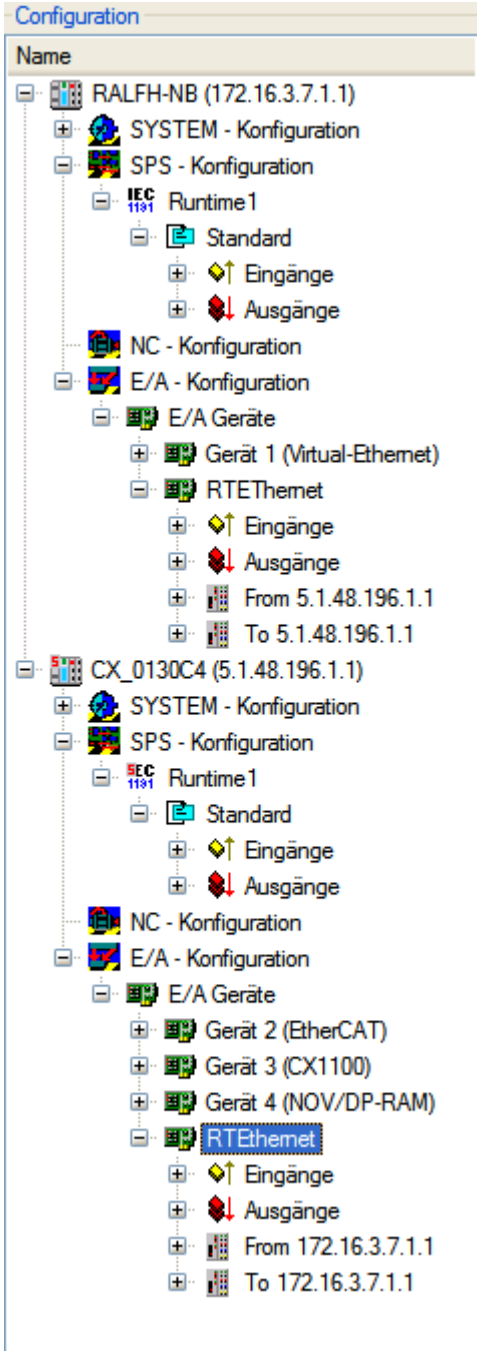


Fig. 27: Figure 1: Navigation window of the TwinCAT Simulation Manager

All shown items that are bound to simulation settings (e.g. Variables / Symbols, Axes, Devices, PLC projects, ...) are marked with a red 'S' in their icon.

The different functionalities of the TwinCAT Simulation Manager are triggered mainly via the 'Navigation Window' context menu. Depending in the selected item in this view, the context menu can contain the commands shown in Table 1.

Requirements

Menu entry	Click context	Description
Create Simulation PLC-Project	Original PLC selected, Simulation PLC not assigned	A Template Simulation PLC project is created and bound to the selected Original PLC (see Create Simulation Project [▶ 31]).
Add Simulation Variables ...	Original PLC selected, Simulation PLC assigned	Maintains new Symbols of the Original PLC into the Simulation PLC (see Add Simulation Symbols [▶ 34]).
Download PLC Symbols	PLC selected	Downloads the *.tpy Symbol information from the target and store them in the local cache of the simulation project. Internal data will be updated.
Import PLC Project	PLC selected	Imports an "external" PLC Control project for local (cached) access. Be sure that the PlcControl project is the actual project running on the target!
Open PlcControl	PLC selected	Opens the PLCControl application for the specified PLC. Precondition is, that the *.pro file was created by the simulation manager (via the Create Simulation Project [▶ 31] command), or was imported by the Import PLC Project
Rescan	PLC selected	Rescans the symbols of the PLC project
Ping	Target System selected	Pings the target system with a UDP-message. If the target is connected to the same subnet and is running, the Ping will succeed.
AdsPing	Target System selected	Pings the target system via ADS. In addition that the target is reachable via UDP, both systems must run a TwinCAT environment (in 'Run' or 'Config' mode) and both systems must refer each other via ADS Routes.
Open System Manager	Target System Selected	Opens the System Manager on the locally cached configuration (*.tsm) downloaded from the target. This configuration should be always the actual configuration of the target.
Download current configuration	Target System Selected	Downloads and caches the 'Current Configuration' from the target.
PlcExplorer	Taget System Selected	Shows a dialog-browser showing the current PLC runtimes of the target.
WebConfiguration	Target System Selected	Opens the WebConfiguration HTML page of the target. If this service is not supported and/or the Web-Server is not running on the target, an error message will be displayed.
Disable	Device(s) Selected	Disables the Device(s) for the simulation profile
Enable	Device(s) Selected	Enables the Devices(s) for the simulation profile
Simulation	Axis Selected	Decouples the Axis in Simulation Mode and sets an 'Simulation Encoder'.
Properties	All	Opens the Properties dialog.

Table 1: Context Menu commands

A few examples for the Context Menu on the most important items for the Simulation project are given in the following.

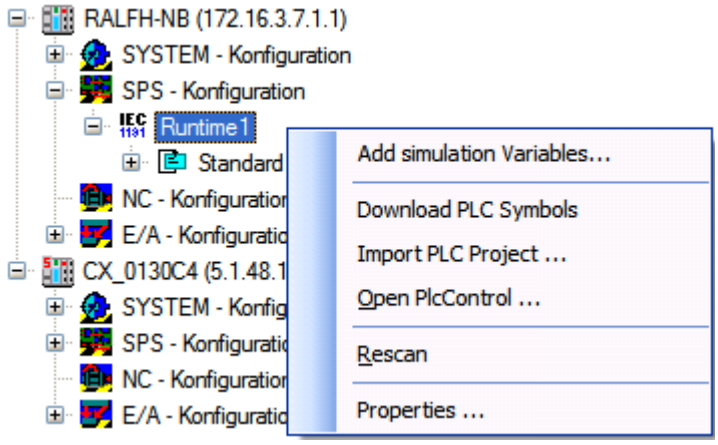


Fig. 28: Figure 2: Context Menu on PLC Runtime when a simulation PLC is already set

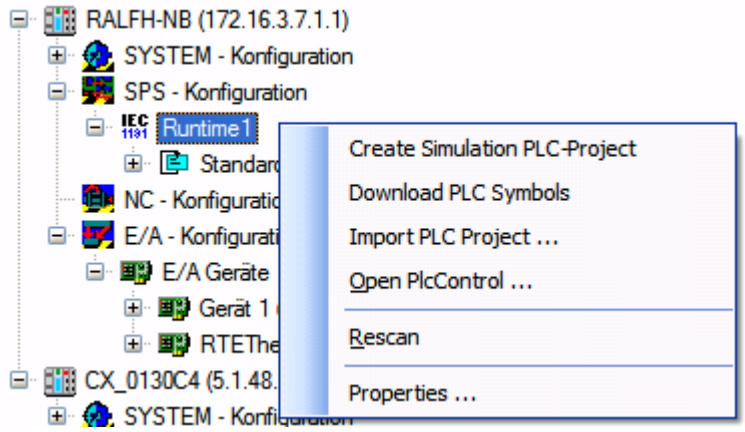


Fig. 29: Figure 3: Context Menu on PLC Runtime if not assigned to a Simulation PLC

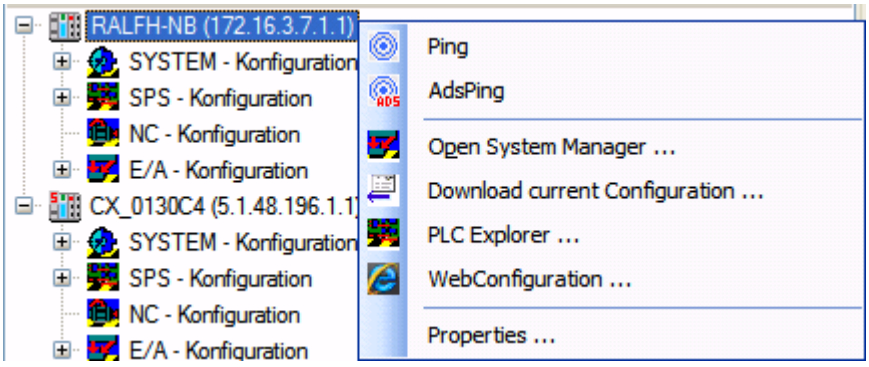


Fig. 30: Figure 4: Context Menu on Target System

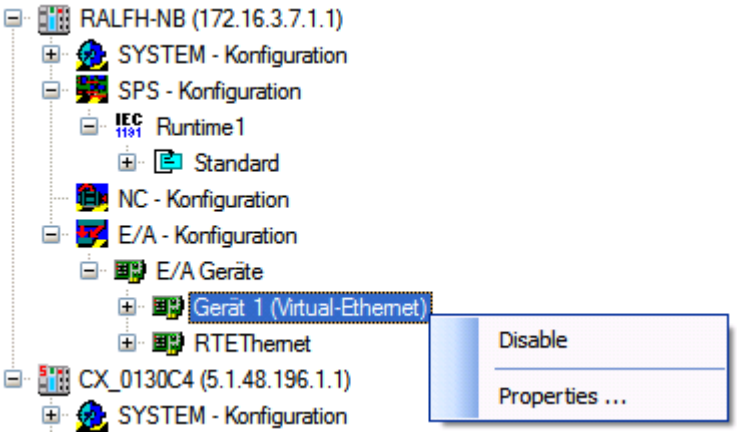


Fig. 31: Figure 5: Context Menu on Device

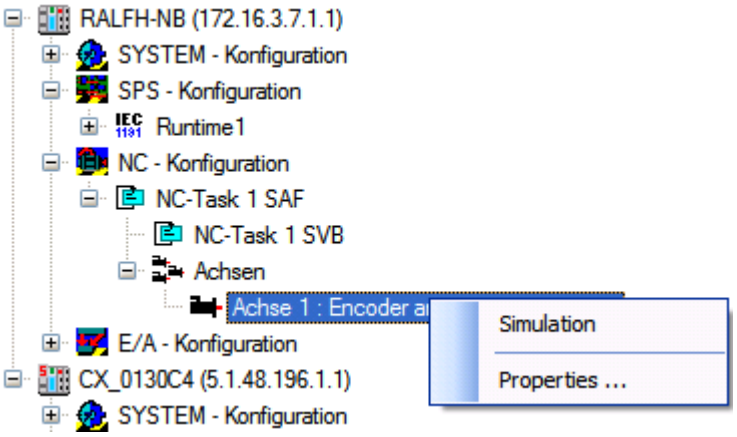


Fig. 32: Figure 6: Context Menu on Axis

5.1.3 System information

The "Project information dialog" displays information about the applied runtime environments. When the "Simulation flag" is set in the PLC, then the corresponding components are marked in yellow.

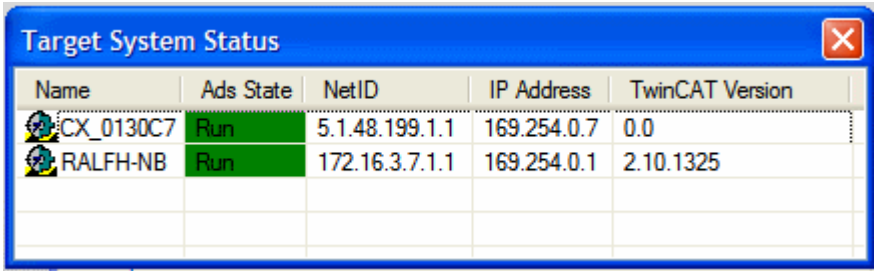


Fig. 33: Figure 1: Target Device State Dialog

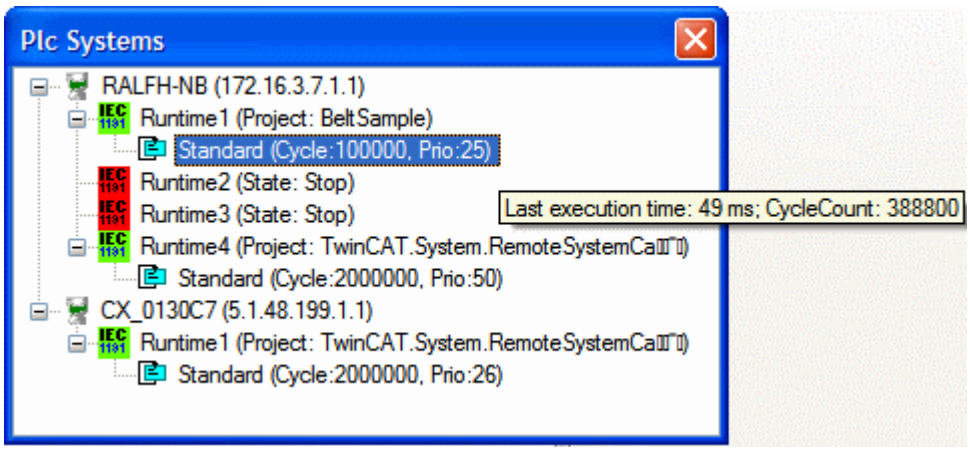


Fig. 34: Figure 1: PLC Status Dialog

5.1.4 Preferences dialog

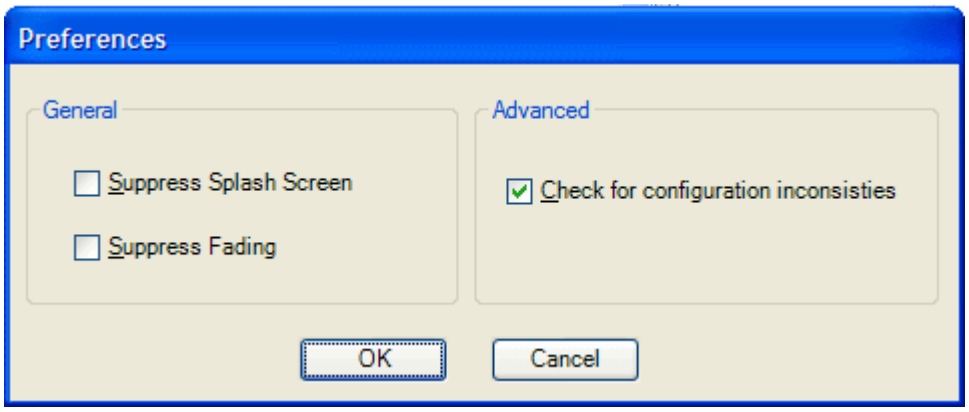


Fig. 35: Figure 1: The 'Preferences' dialog

Requirements

Option	Description
Check for configuration inconsistencies	In TwinCAT System Manager configuration files (*.wsm or *.tsm), which have been created with a TwinCAT Version before <2.9, cannot provide unique node names. This means that a given node name is allocated twice in the same sub branch. Configurations created with Version 2.9 no longer contain this ambiguity. To ensure perfect functioning of the TwinCAT Simulation Manager, the inconsistencies must be eliminated. To this end, a testing mechanism is provided, which examines the system manager configuration when a new simulation manager project is created and if appropriate issues a corresponding error message. This test is complex and takes a certain amount of time, it can therefore be switched off. However this should only be done if it has already been ensured that there are no more inconsistencies in the configurations used. For newer TwinCAT System Manager configurations, this check can be checked off.
Suppress Splash Screen	Suppresses the Splash Screen
Suppress Fading	Suppresses the Fading of the progress window and the splash screen

Table 1: Options of the Preferences Dialog

The Preferences will be persisted as User-settings.

5.1.5 Context dependant

5.1.5.1 Axes view

When an "axis" group node is activated in the navigation tree, a list of axes opens on the right of the screen. These axes can be marked as "simulated" or "original" via the context menu.

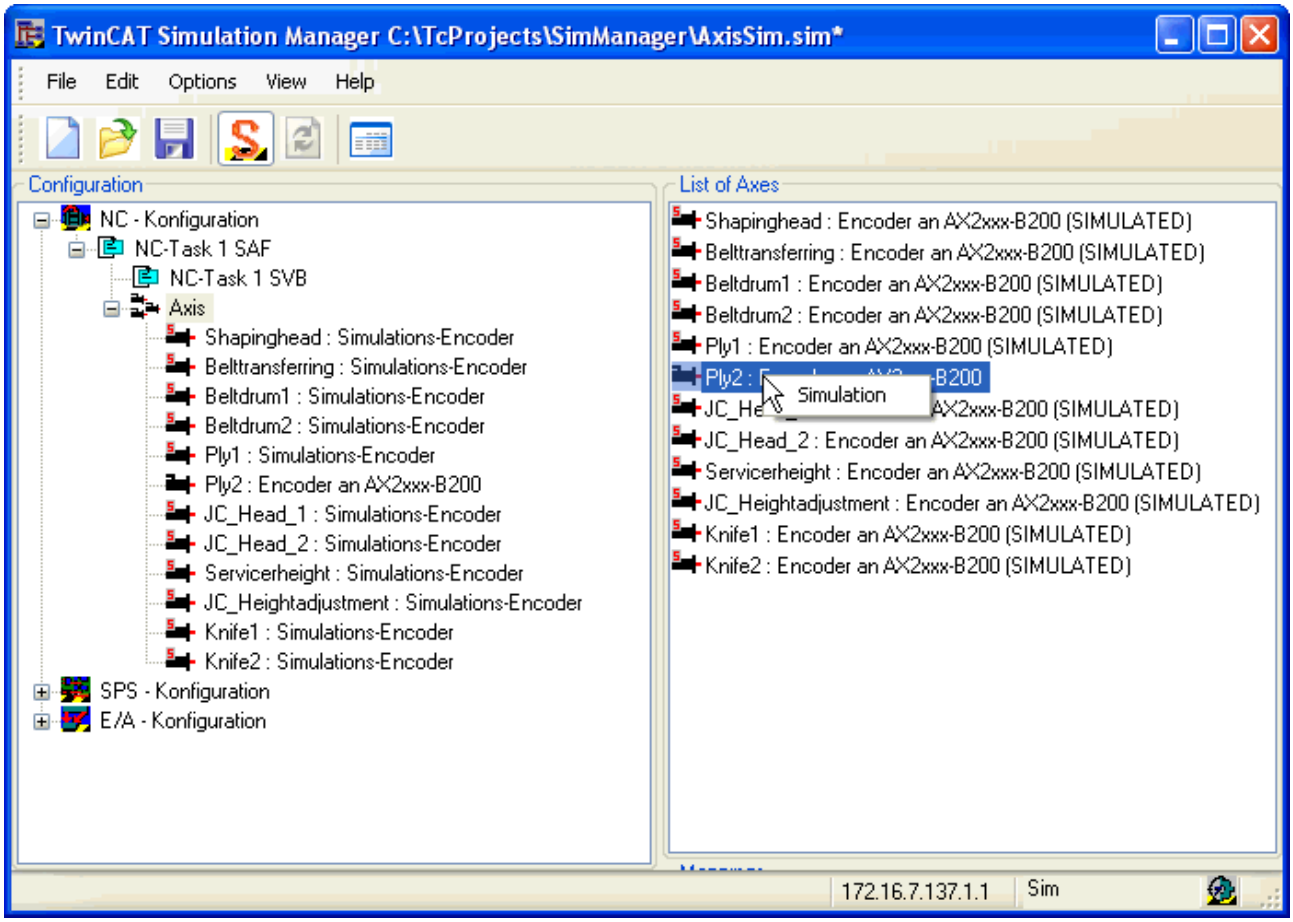


Fig. 36: Figure 1: Axes view

Every simulated axis is labelled as such in the icon.

Name	Description
Simulation	Selects the Axis for Simulation
Original	Deselects the simulation setting for the Axis

Tabelle 1: Contextmenue commands in the "Axes View"

5.1.5.2 Axes view

The axis view displays the online data of the axes. When the axis is in simulation mode, the operating elements for "jogging" the Axis are active.

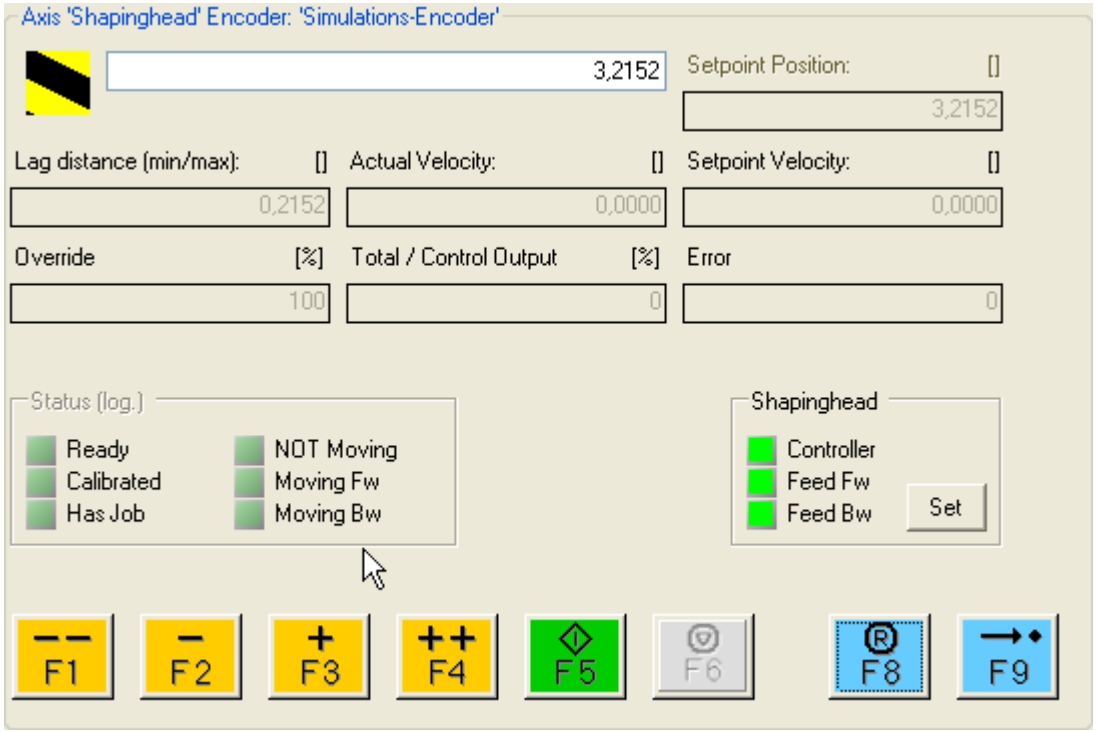


Fig. 37: Figure 1: Axis in simulation mode

5.1.5.3 Simulation Symbols Configuration View

The window "Simulation variables" shows the current and selectable simulation bindings for a variable symbol for one in the Navigation window [▶ 39] selected symbol variable. If a symbol of an Original-PLC is selected, only simulation PLC variables appear in the selection list. If a simulation variable is marked, then only Original-Symbols appear. This reversal is highlighted by a new window name (instead of "Simulation variables," "Original PLC variables" appear)

Only symbols of the same data type with reversed allocations can be selected (by checkbox) (inputs only with outputs, or outputs with inputs).

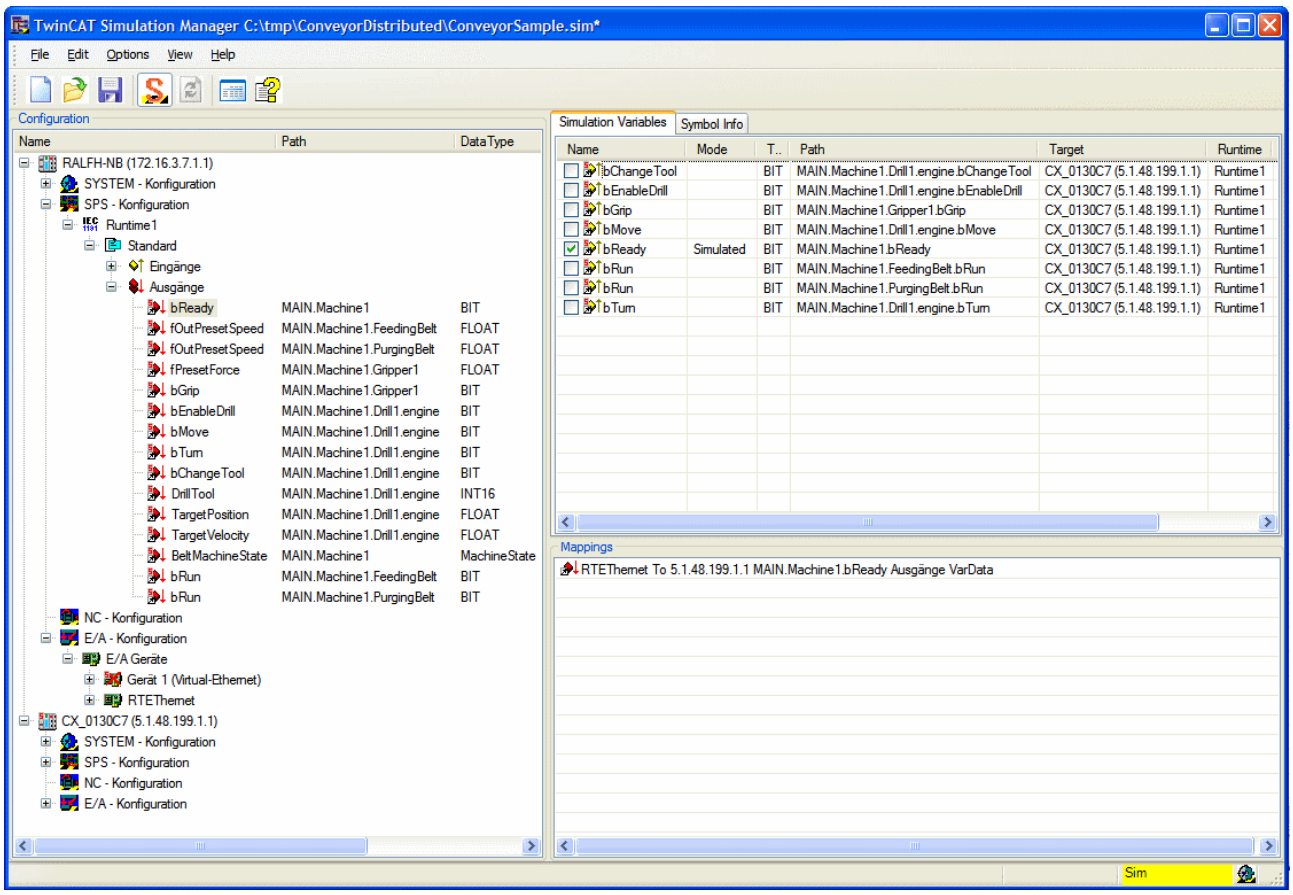


Fig. 38: Figure 1: Selected variables symbol in the navigation window and "Simulation variables" screen

Any Standard mappings [▶ 35] that are identified will be highlighted in green.

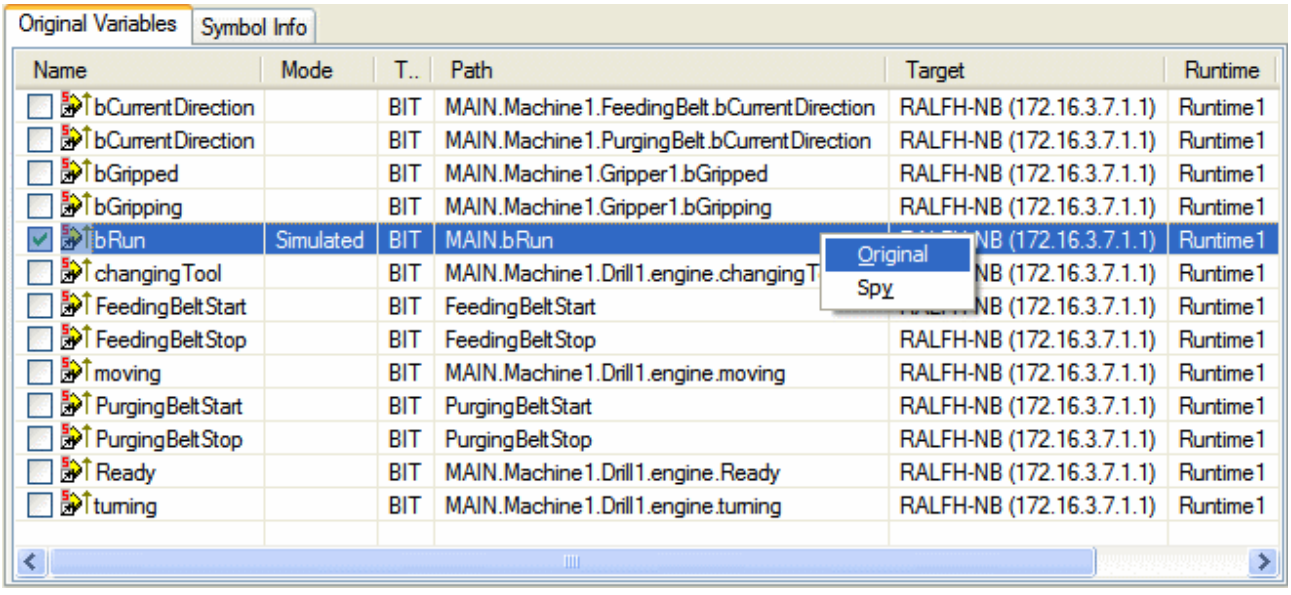


Fig. 39: Figure 2: Setting the simulation mappings

The simulation settings can be changed as follows via checkboxes in the view window or via the context menu:

Type of mapping	Description
Observe	Observation of an I/O
Simulate	Simulation of an I/O, the mapping and the device are removed in simulation mode.

Type of mapping	Description
Original	No change in simulation mode.

Table 1: Context Menu Commands of the 'Simulation Symbols Configuration View'

Variables or symbols that are connected differently in simulation mode, are indicated with a red S in the top-left corner of their icon.

5.1.5.4 Current Mappings View

The mapping window shows the active mappings for the variable selected in the navigation window [▶ 39]. Dependant on the profile (Original or Simulation) and the simulation type (local or remote) the Symbol can be bound to IO-Devices, to other PLCs or to Network Publisher / Subscriber Variables (configured by the Simulation Symbols Configuration View [▶ 45])

Example:

The activated profile is the 'Original' mode. Within this profile the symbol '**MAIN.bRun**' of the Original-PLC is bound to an Input-Terminal on an BK9000 device (Figure 1).

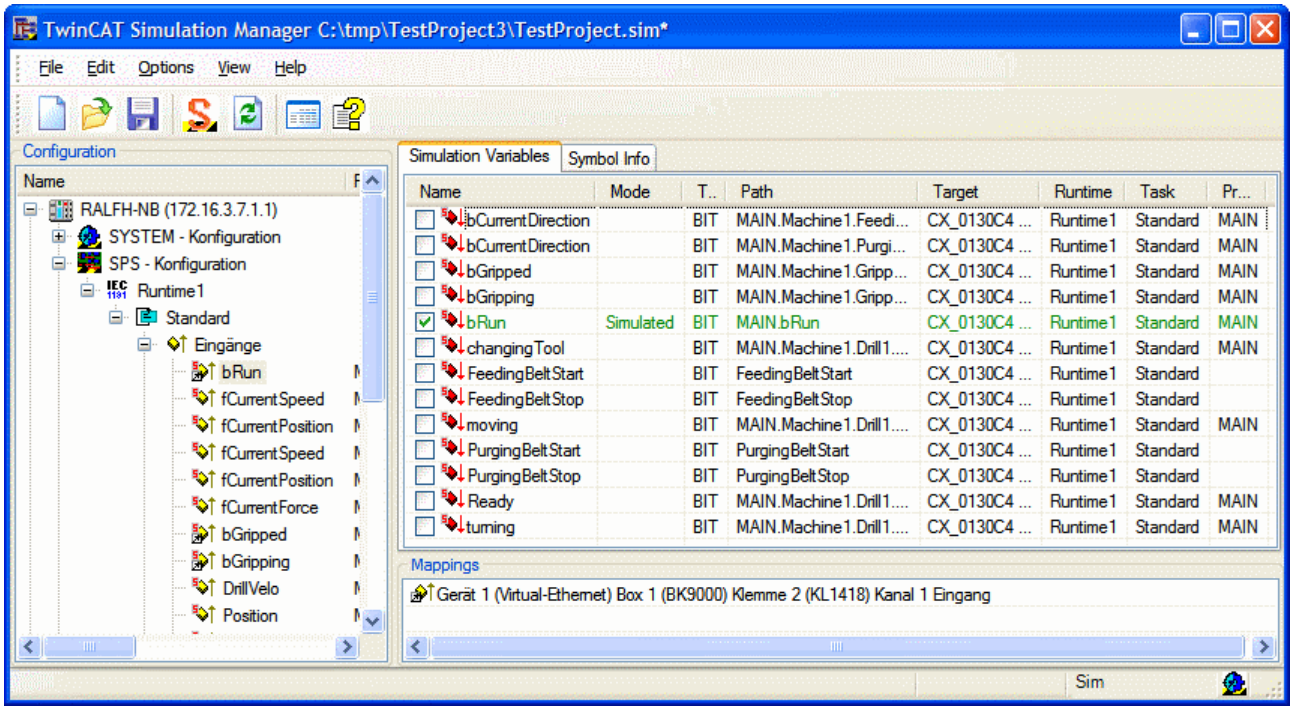


Fig. 40: Figure 1: IO Linked in 'Original' profile

In the Simulation-Profile the Symbol '**MAIN.bRun**' is linked to a network subscriber variable named 'MAIN.bRun' (see Figure 2). The network variable gets its data from a system with the address '5.1.48.196.1.1'.

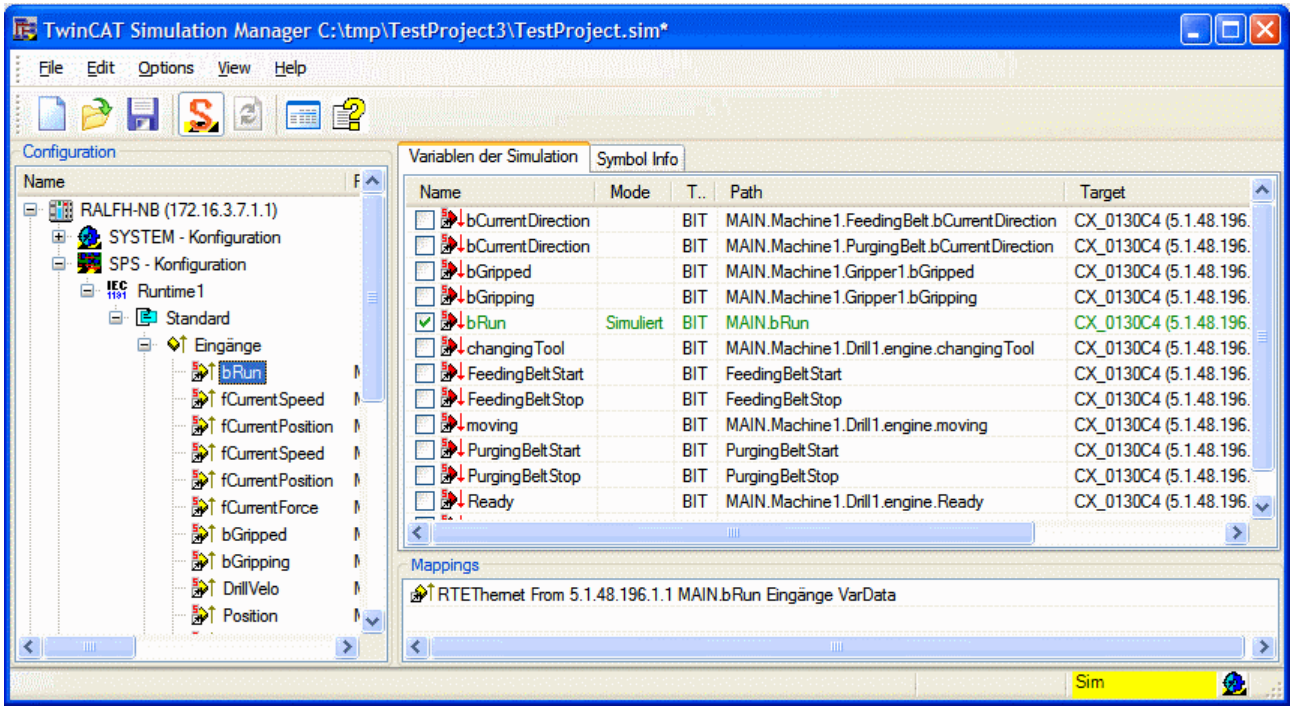


Fig. 41: Figure 2: Network variable Subscriber linked in Simulation profile

5.2 Main menue structure

5.2.1 Main menu "File"

Menu entry	Description
New	Creates a new simulation manager project
Open	Opens an existing simulation manager project
Close	Closes the currently open simulation manager project
Save	Saves the currently open simulation manager project
Save As	Saves the currently open simulation manager project under another name
Target Device State	Opens the Target Device Controller [▶ 42] Dialog. This shows the status of the configured target systems and enables the user to Start / Stop these targets.
PLC State	Opens the PLC Status [▶ 42] Dialog. This dialog shows the status of the used PLC runtimes and their task information.
Exit	Exits the simulation manager

Table 1: Commands of the 'File' Main menu

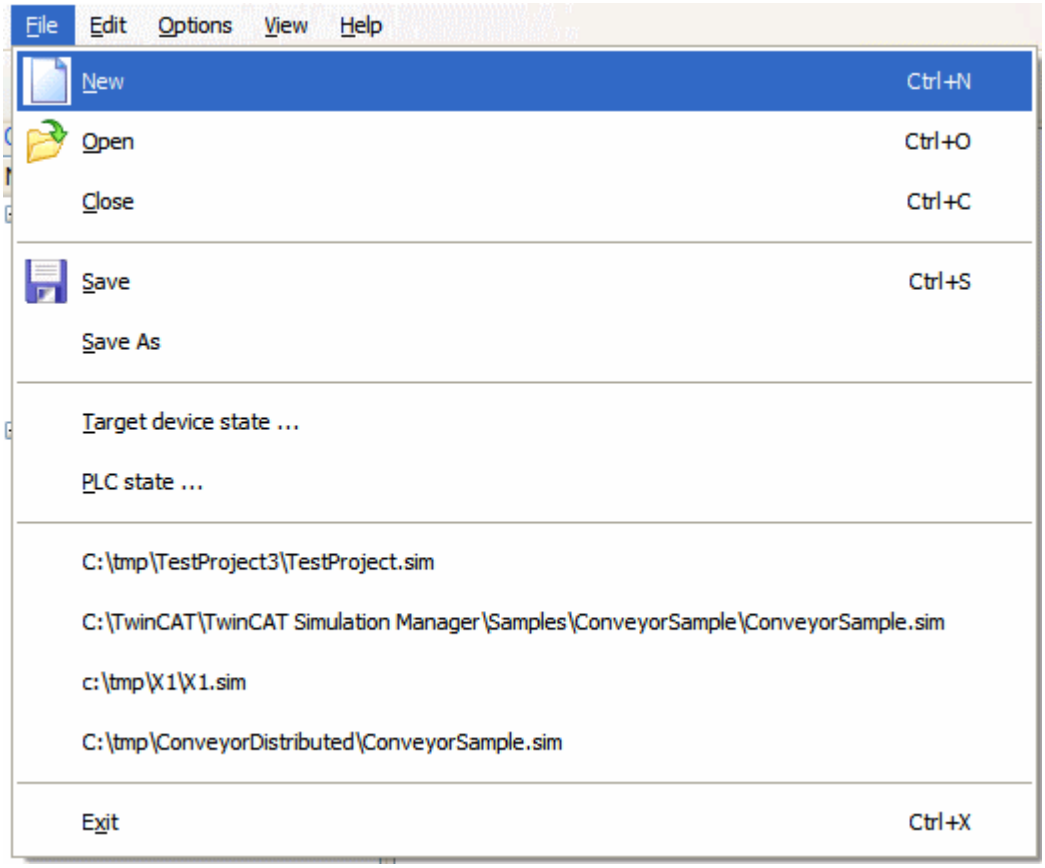


Fig. 42: Figure 1: The 'File' Main menu

5.2.2 Main menu "Edit"

Menu entry	Description
Simulation Mode	Toggles the system between Simulation Profile [▶ 25] and Original Profile [▶ 28].
Overwrite device configuration ...	Sets the current profile again. All corresponding settings that are part of the TwinCAT Simulation manager configuration project will be reassigned.
PLC Project mappings ...	Opens the dialog for maintaining the Original PLC to Simulation PLC assignments [▶ 25].
Write simulation state to PLCs	Writes the simulation flag into all configured PLCs (see PLC simulation library TcSimMan.lib [▶ 57]). The **dwSimState** flag will be set to the appropriate state (0: for original profile, 1: for simulation profile)
Add simulation variables...	Opens the "Add Symbols to Simulation" Dialog [▶ 34]. This function is used for maintaining subsequently added Original-PLC symbols into the Simulation-PLC.
Select target Systems ...	Opens the Dialog to Add / Remove target systems [▶ 20].
Create default mappings	Creates the standard bindings between Original-PLCs and Simulation-PLCs for the current project (see Default Mappings [▶ 35]). Eventually predefined bindings will be deleted.
Reset Simulation settings	All simulation settings will be reset.
Check Configuration	Checks the TwinCAT Simulation Manager configuration for consistency.
Rescan PLC projects	The symbols of all PLCs are re-loaded.

Table 1: Commands within the 'Edit' Main menue

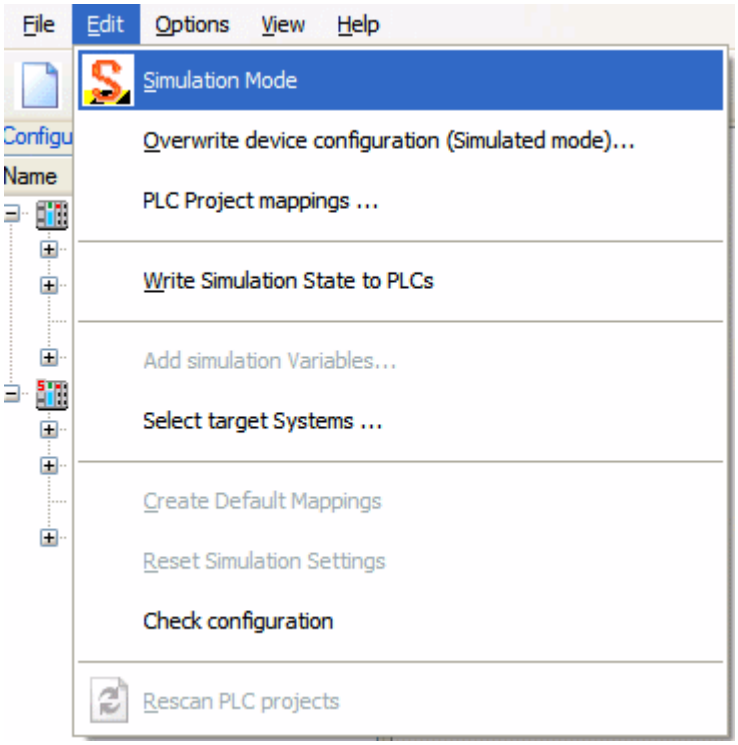


Fig. 43: Figure 1: The 'Edit' Main menue

5.2.3 Main menu "Options"

Menu entry	Description
Languages	Language settings (user dependent) Supported Languages: • English • German
Preferences	Opens the TwinCAT Simulation Manager Preferences [▶ 43] Dialog

Table 1: Commands of the 'Options' main menue

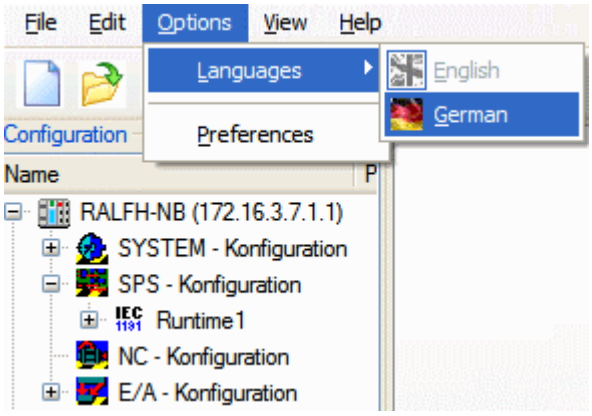


Fig. 44: Figure 1: The 'Options' main menue

5.2.4 Main menu 'View'

Menu entry	Description
Actions	Opens the Simulation Actions Dialog [▶ 51].
References	Opens the Overview Dialog for Actual Mappings [▶ 52]

Table 1: Commands of the 'View' main menu

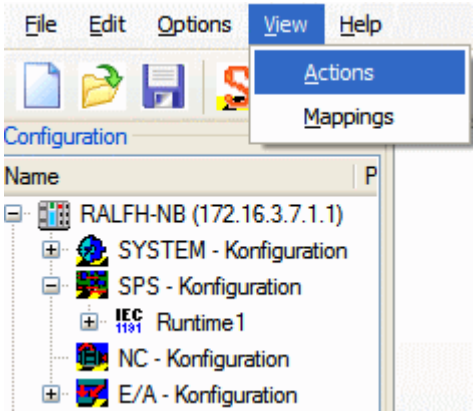


Fig. 45: Figure 1: The 'View' Main menu

5.3 Dialogs

5.3.1 Dialog 'Simulation Actions'

The 'Simulation Actions' Dailog shows the Configuration-Actions, that are necessary to switch the TwinCAT Simulation Manager Profile form from 'Original' to 'Simulation'.

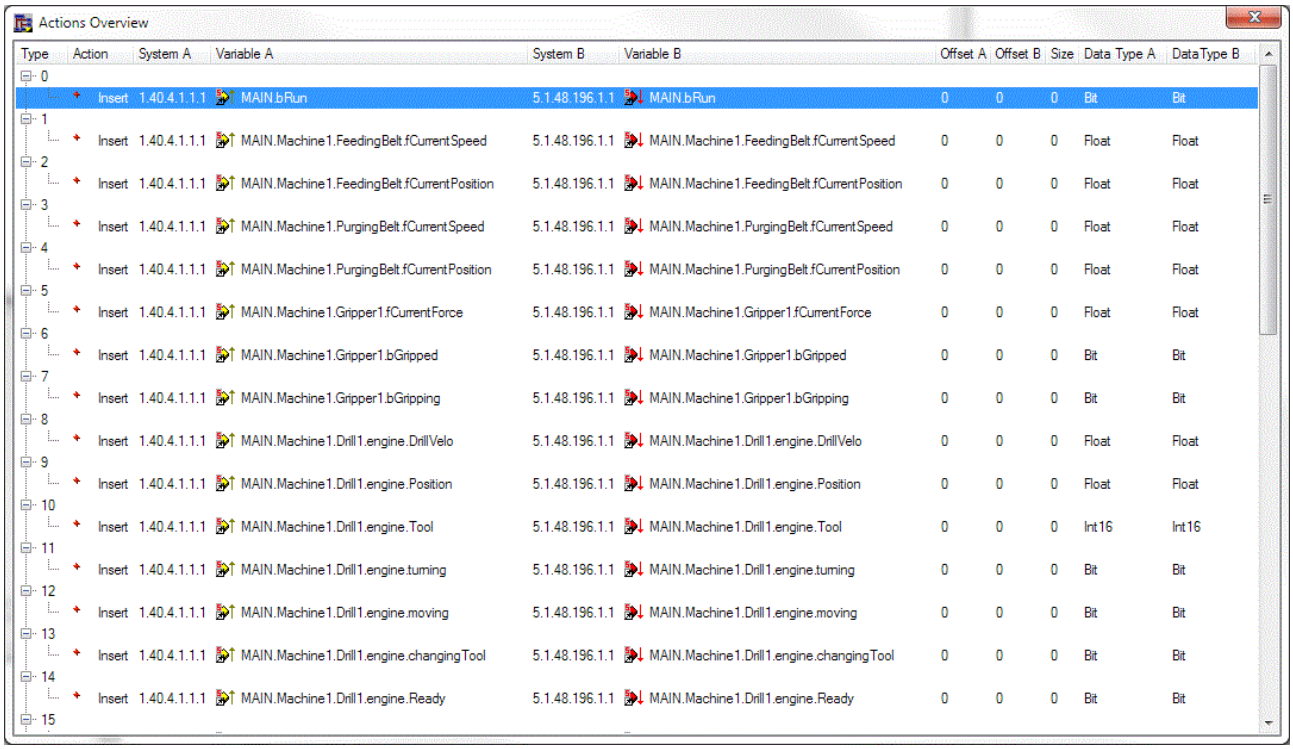
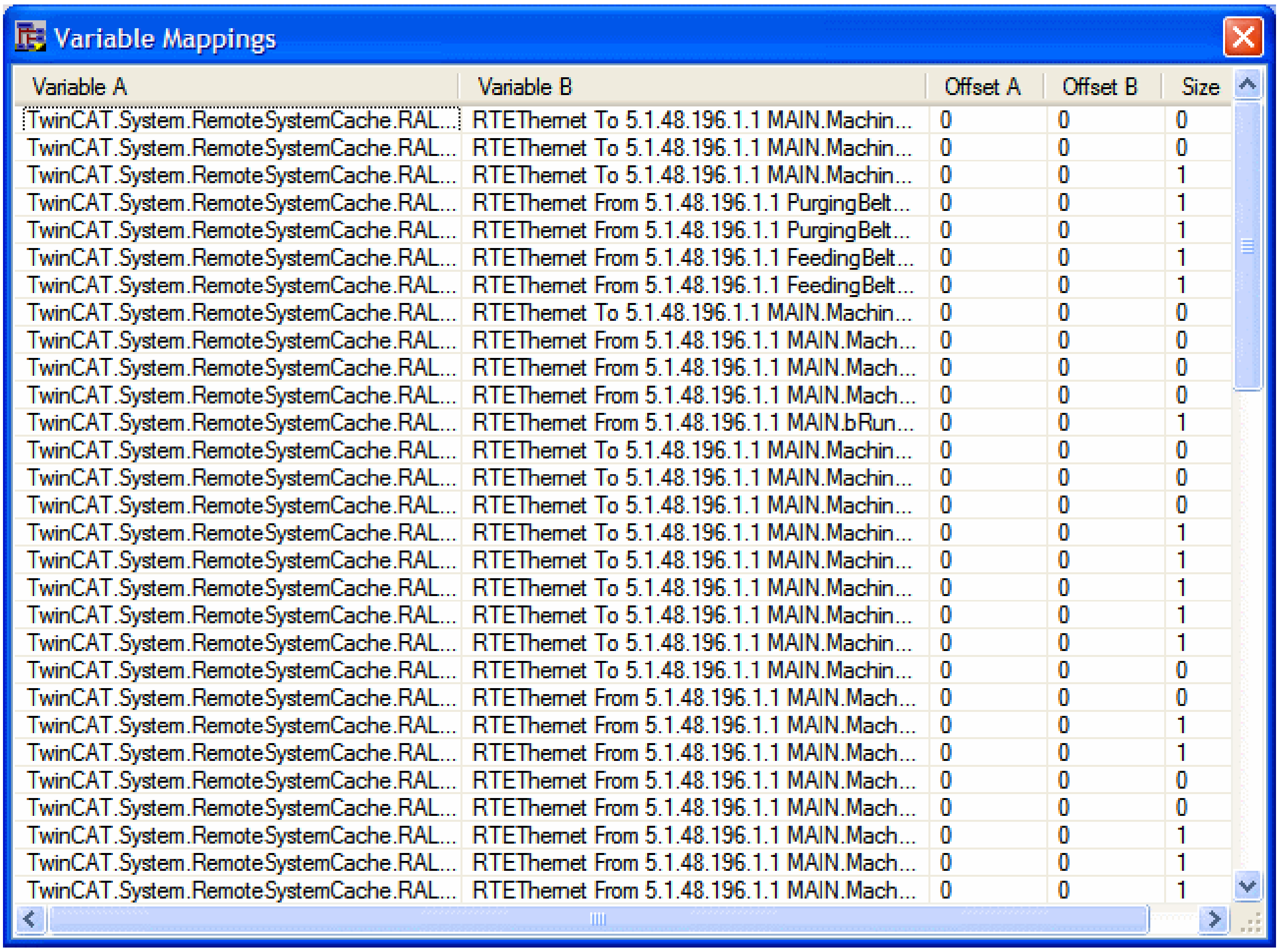


Fig. 46: Figure 1: The 'Simulation Actions' Dialog

5.3.2 Dialog 'Overview actual mappings'

The dialog 'Overview of variable mappings' shows the set mappings of all target systems.



Variable A	Variable B	Offset A	Offset B	Size
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 PurgingBelt...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 PurgingBelt...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 FeedingBelt...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 FeedingBelt...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.bRun...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet To 5.1.48.196.1.1 MAIN.Machin...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	0
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	1
TwinCAT.System.RemoteSystemCache.RAL...	RTEThernet From 5.1.48.196.1.1 MAIN.Mach...	0	0	1

Fig. 47: Figure 1: Overview of target mappgins

5.3.3 Dialog "Broadcast Search"

The Dialog 'Broadcast Search' helps to find TwinCAT target systems within the intranet and to select them.

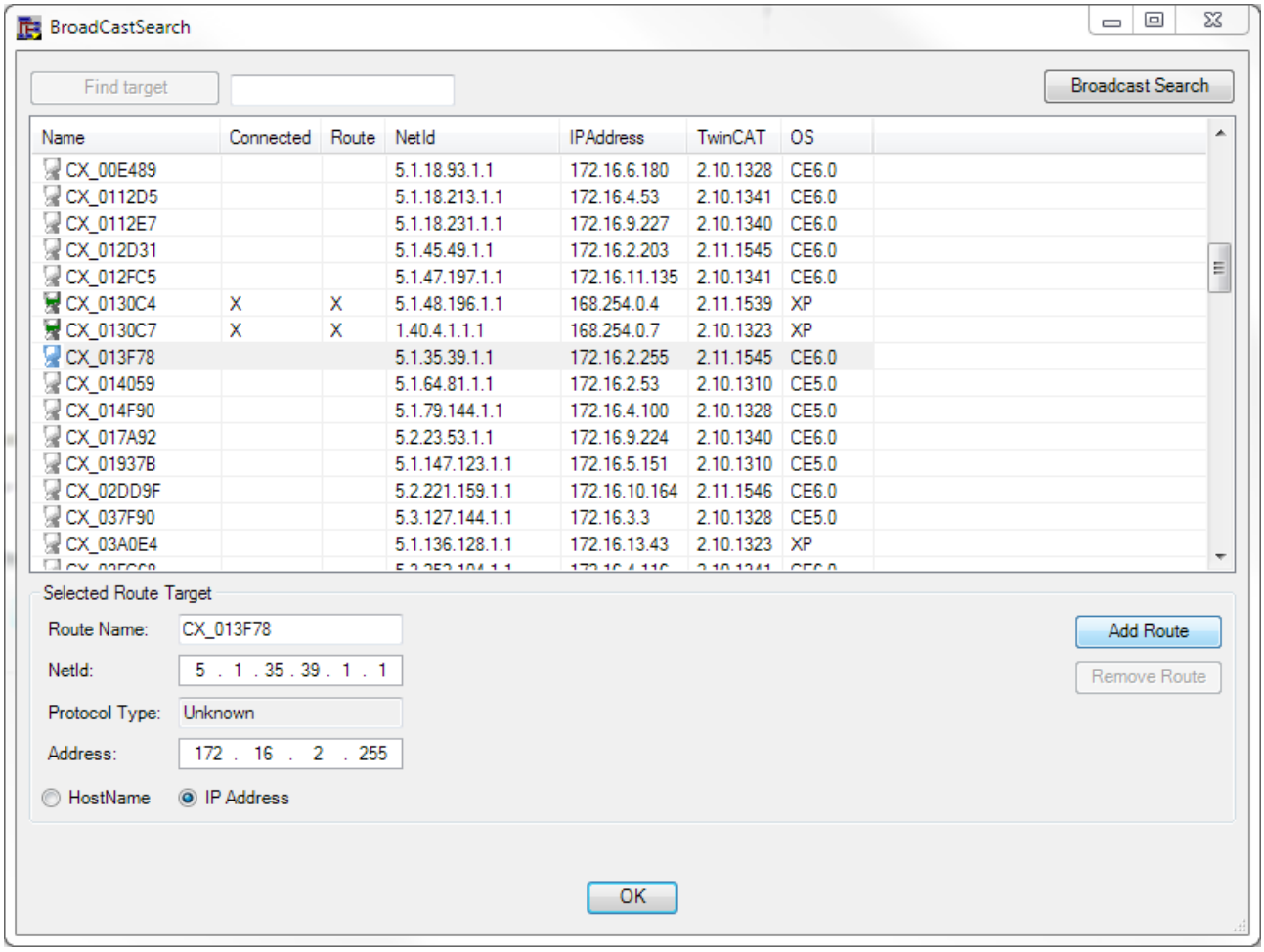


Fig. 48: Figure 1: The Dialog 'Broadcast Search'

Fuction	Description
Broadcast Search	Searches for all TwinCAT Devices within the network and shows them in the listview.
Find target	Searches for a TwinCAT Device with a specific Name (HostName), AmsNetId or IPAddress.
Add route	Adds the selected target system as (remote) route to the local system (and vice versa).
Remove Route	Removes the selected target system from the local system (and vice versa).
OK	Closes the dialog.

Table 1: Functions of the Dialog 'Broadcast search'

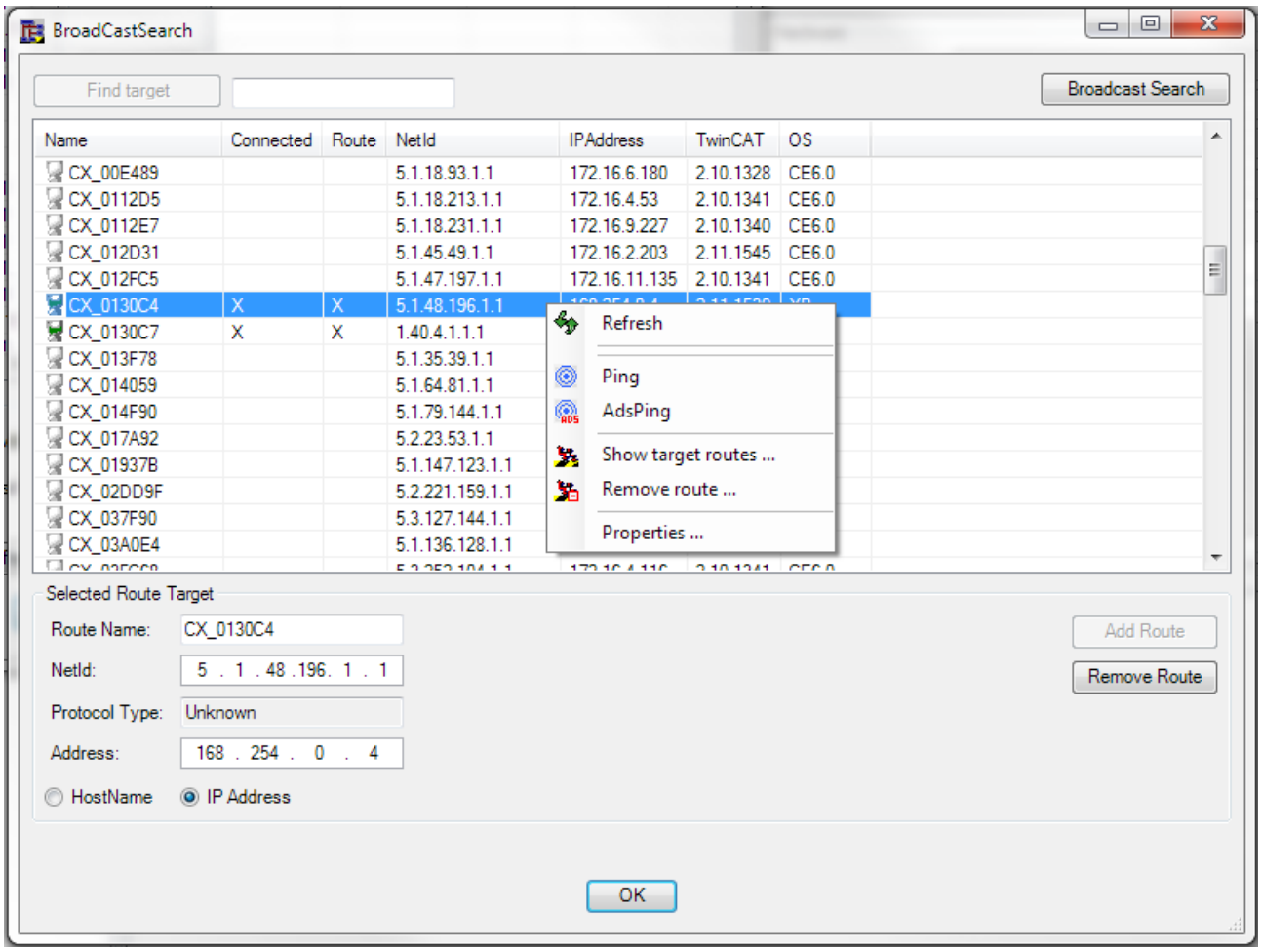


Fig. 49: Figure 2: Context menu of the Dialog 'Broadcast search'

Menu entry	Description
Refresh	Sends an immediate broadcast search message and aktualizes the list of target systems.
Ping	Pings the target system with a UDP-message. If the target is connected to the same subnet and is running, the Ping will succeed.
AdsPing	Pings the target system via ADS. In addition that the target is reachable via UDP, both systems must run a TwinCAT environment (in 'Run' or 'Config' mode) and both systems must refer each other via ADS Routes. This Context Menu Item is only active when the target system is assigned as local route.
Show target routes ...	Show the routes that are registered at the selected target (only if the target route is assigned to the local system)
Add route ...	dds the selected target system as (remote) route to the local system (and vice versa).
Remove route ...	Removes the selected target system from the local system (and vice versa).
Properties ...	Shows the properties of the selected target system.

Table 2: Context menu of the Dialog 'Broadcast search'

5.3.4 Dialog 'Manage target systems'

The dialog 'Manage target systems' is used to manage the target systems within the actual **TwinCAT Simulation Manager** project.

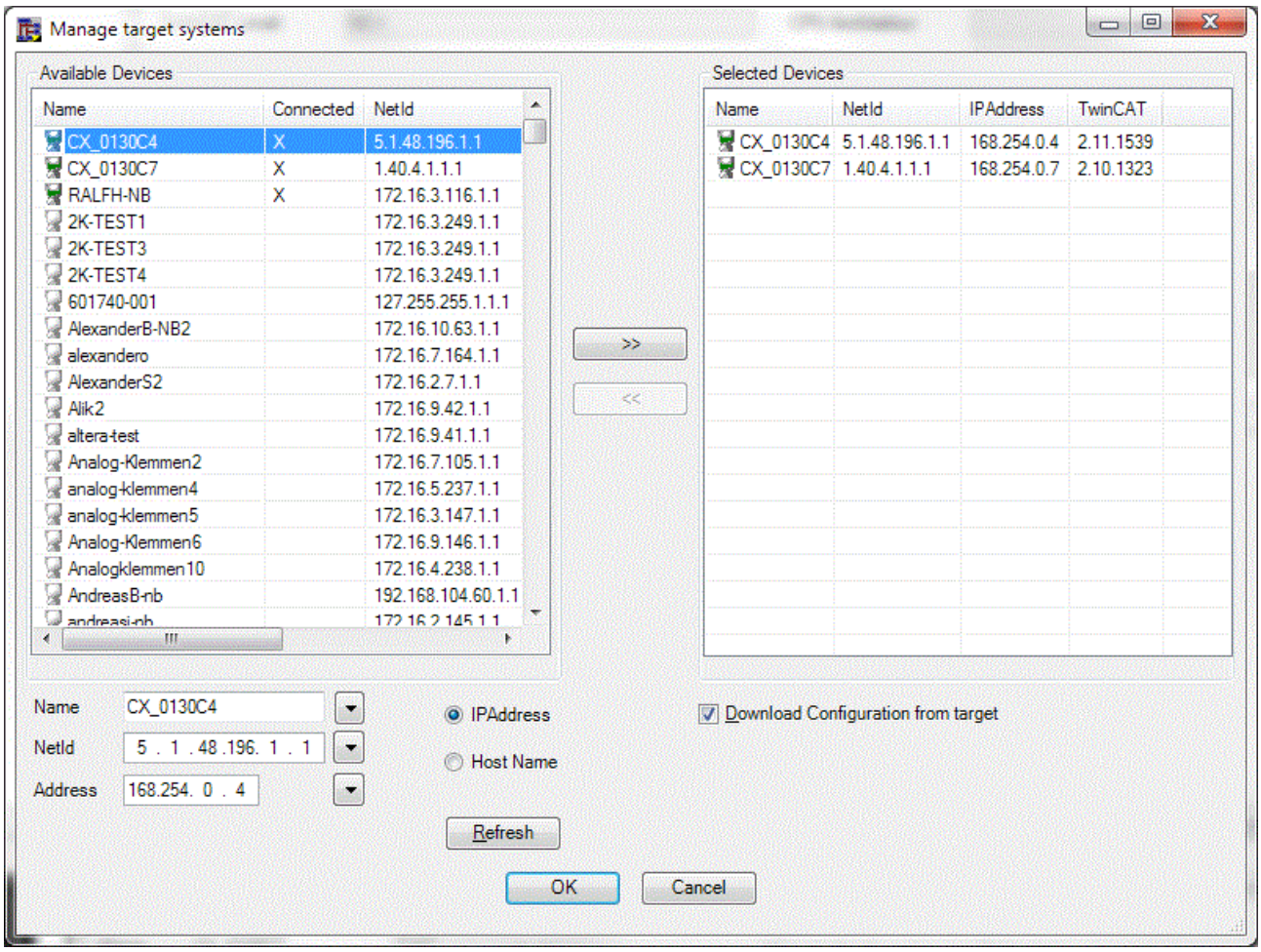


Fig. 50: Figure 1: Dialog 'Manage target systems'

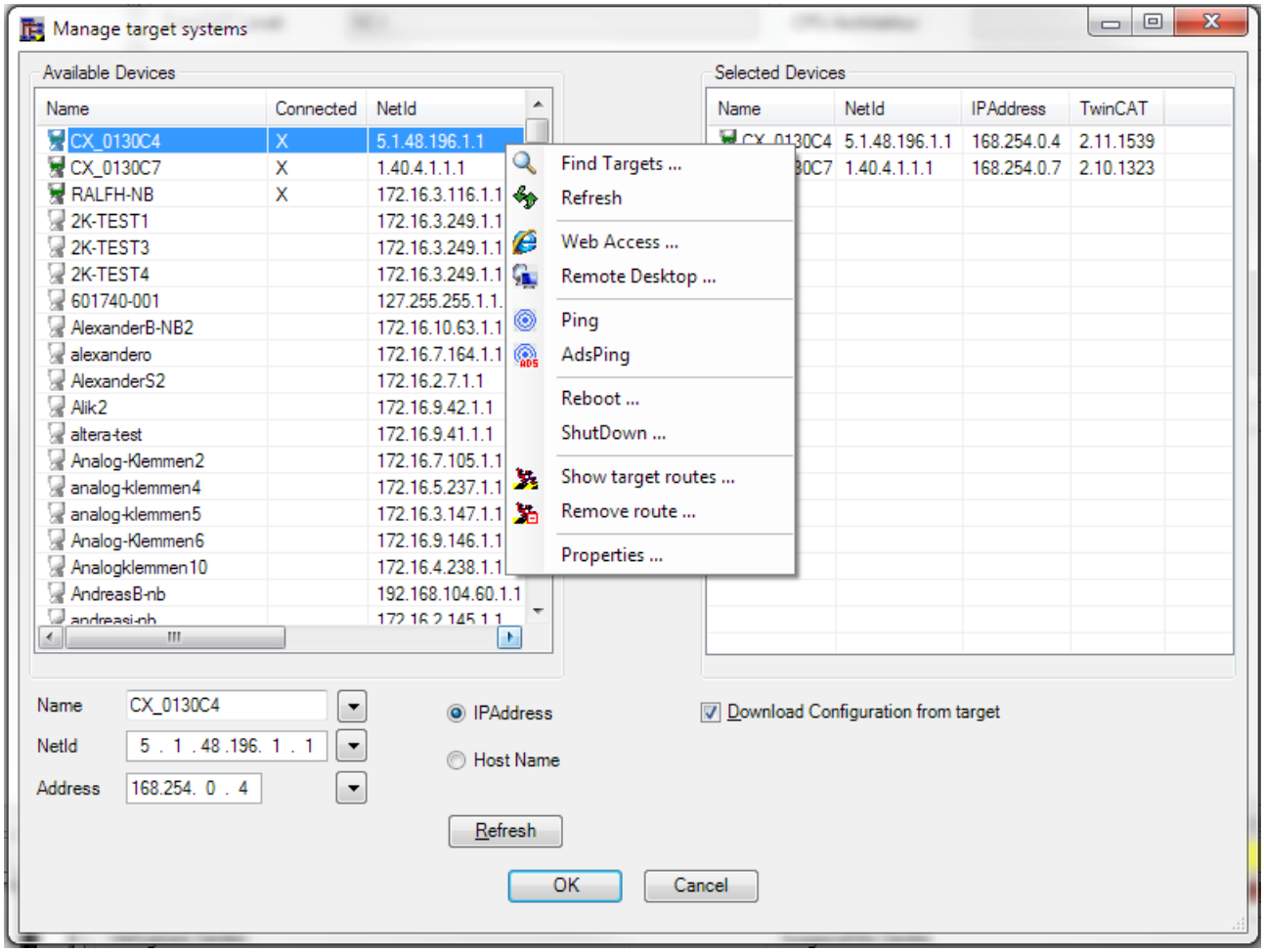


Fig. 51: Figure 2: Context Menue Dialog 'Manage target systems'

Menu entry	Description
Find targets ...	Searches for target systems and assigns ADS routes.
Refresh	Sends a new broadcast search immediately and updates the list
Web Access ...	Opens the WebConfiguration HTML page of the target. If this service is not supported and/or the Web-Server is not running on the target, an error message will be displayed.
Remote Desktop ...	Opens a remote Desktop dialog showing the desktop of the target system.
Ping	Pings the target system with a UDP-message. If the target is connected to the same subnet and is running, the Ping will succeed.
AdsPing	Pings the target system via ADS. In addition that the target is reachable via UDP, both systems must run a TwinCAT environment (in 'Run' or 'Config' mode) and both systems must refer each other via ADS Routes. This Context Menu Item is only active when the target system is assigned as local route.
Reboot ...	Reboots the target system.
ShutDown ...	Shutdown of the target system.
Show target routes ...	Show the routes that are registered at the selected target (only if the target route is assigned to the local system)
Remove route ...	Removes the route.
Properties ...	Target properties

Table 1: Context Menue Dialog 'Manage target systems'

6 PLC API

If a PLC is to be simulated via the TwinCAT Simulation Manager, the original PLCs and the simulation PLC must contain the TcSimManager.lib.

Global Variables

TcSimManager.lib contains the universal DWORD 'dwSimState', through which the PLCs are informed about the current state of the simulation.

```
VAR_GLOBAL
    dwSimState : DWORD;
END_VAR
```

Flag	Description
0	Simulation mode on / off The flag is set by the TwinCAT Simulation Manager when the switchover to the simulation mode has been successful. At this point the mode can be read by the PLC.
1 - 31	Reserved

Blocks

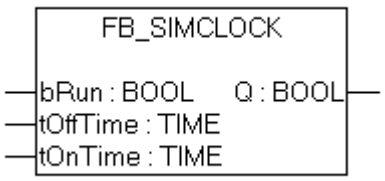
Name	Description
FB_SimDelayedLatch [▶ 59]	Delayed latch
FB_SimPosMotor [▶ 63]	Position-controlled simulation motor
FB_SimSimpleVeloMotor [▶ 64]	Simple velocity-controlled simulation motor
FB_SimVeloMotor [▶ 65]	Velocity-controlled simulation motor

Auxiliary blocks

Name	Description
FB_SimClock [▶ 57]	Clock block
FB_SimDelay [▶ 58]	Delay block
FB_SimDerivative [▶ 60]	Derivative block
FB_SimIntegral [▶ 61]	Integral block
FB_SimPastCpuCounterTicks [▶ 62]	Timing block
FB_SimRamp [▶ 60]	Ramp block

6.1 FB_SimClock

Clock block with variable pulse width.



VAR_INPUT

```
VAR_INPUT
    bRun     : BOOL; (* Enabled *)
    tOffTime : TIME; (* Down Time of a pulse *)
    tOnTime  : TIME; (* Up Time of a pulse *)
END_VAR
```

bRun: Activates the clock block

tOffTime: Duration of low level

tOnTime: Duration of high level

VAR_OUTPUT

```
VAR_OUTPUT
    Q        : BOOL; (* PULSE *)
END_VAR
```

Q: Output signal

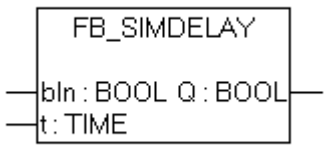
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.2 FB_SimDelay

Delay block.

A binary input signal 'bIn' is issued for the time duration 't'. The delay must be shorter than the input pulse duration.



VAR_INPUT

```
VAR_INPUT
    bIn  : BOOL;  (* Input pulse *)
    t    : TIME; (* Delay time of the pulse *)
END_VAR
```

bIn: Input signal

t: Delay time (should be shorter than the pulse duration)

VAR_OUTPUT

```
VAR_OUTPUT
    Q    : BOOL; (* PULSE *)
END_VAR
```

Q: Output signal

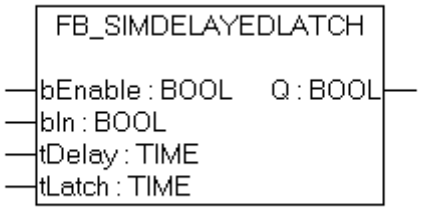
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.3 FB_SimDelayedLatch

Delayed latch

The output '**Q**' is delayed with respect to the input '**bIn**' by a certain time 'tDelay'. An output 'TRUE'-level for a time '**tLatch**' is maintained.



,

VAR_INPUT

```
VAR_INPUT
    bEnable   : BOOL;    (* Activates the Latch *)
    bIn       : BOOL;    (* Input signal *)
    tDelay    : TIME;    (* Delay of the output signal *)
    tLatch    : TIME;    (* Latch time of the delayed output signal
*)END_VAR
```

bEnable: Activates the block

bIn: Input signal

tDelay: Delay of output signal

tLatch: Latch time of TRUE output level

VAR_OUTPUT

```
VAR_OUTPUT
    Q     : BOOL;    (* Output signal *)
END_VAR
```

Q: Output signal

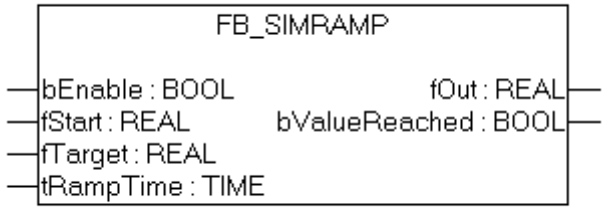
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.4 FB_SimRamp

Ramp block

A signal that changes linearly over time can be simulated via the ramp block. The signal initialises with the value '**fStart**' and ends after a time '**tRampTime**' with the value '**fTarget**'.



VAR_INPUT

```
VAR_INPUT
    bEnable     : BOOL; (* Starts ramp with rising edge, Stops ramp with falling edge *)
    fStart      : REAL; (* Start value *)
    fTarget     : REAL; (* Target value *)
    tRampTime   : TIME; (* Time needed from fStart to fTarget *)
END_VAR
```

bEnable: The ramp is started with a rising edge, and stopped with a falling one

fStart: Start value

fTarget: Target value

tRampTime: Time duration from fStart to fTarget

VAR_OUTPUT

```
VAR_OUTPUT
    fOut          : REAL; (* Output value *)
    bValueReached : BOOL; (* Target value reached *)
END_VAR
```

fOut: Current value

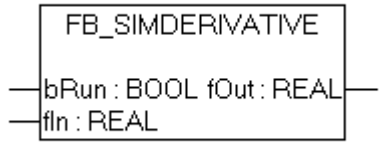
bValueReached: TRUE : Target value reached

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.5 FB_SimDerivative

Derivative block



VAR_INPUT

```
VAR_INPUT
    bRun      : BOOL; (* 0 = Reset *)
    fIn       : REAL; (* Input for derivative *)
END_VAR
```

bRun: Activates the block

fIn: Input for derivation

VAR_OUTPUT

```
VAR_OUTPUT
    fOut      : REAL; (* Derivative output *)
END_VAR
```

fOut: Derived value

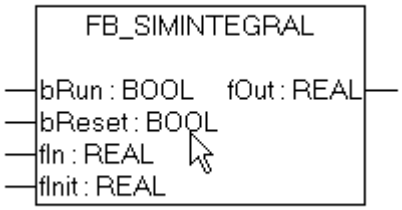
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.6 FB_SimIntegral

Integral block

The integral block integrates the input value 'fIn' with respect to time. The time difference that elapses between two calls of the block serves as the time base.



VAR_INPUT

```
VAR_INPUT
    bRun    : BOOL; (* Integrate 1, Hold 0 *)
    bReset  : BOOL; (* Reset *)
    XIN     : REAL; (* Input for Integral*)
    X0      : REAL; (* Initial value *)
END_VAR
```

bRun: Activates the block

bReset: Initialises the block with the initialisation value in 'fInit'

fIn: Input value for integration

fInit: Initialisation value

VAR_OUTPUT

```
VAR_OUTPUT
    fOut       : REAL; (* Integral out, Integrated over milliseconds*)
END_VAR
```

fOut: Integrated value

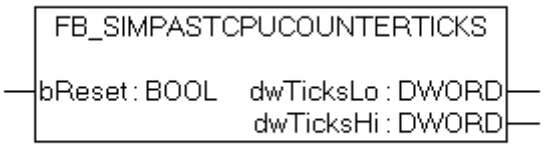
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.7 FB_SimPastCpuCounterTicks

Timing module

This block measures the time that elapses between two block calls in 100 ns ticks.



VAR_INPUT

```
VAR_INPUT
    bReset : BOOL; (* TRUE = > reset cpu counter tick measure*)
END_VAR
```

bReset: Initialisation of timing

VAR_OUTPUT

```
VAR_OUTPUT
    (* one digit = 100 nanoseconds *)
    dwTicksLo : DWORD;    (* Low DWORD of the time delta *)
    dwTicksHi : DWORD;    (* High DWORD of the time delta *)
END_VAR
```

dwTicksLo: Time difference lower DWORD (in 100ns ticks)

dwTicksHi: Time difference upper DWORD (in 100ns ticks)

Requirements

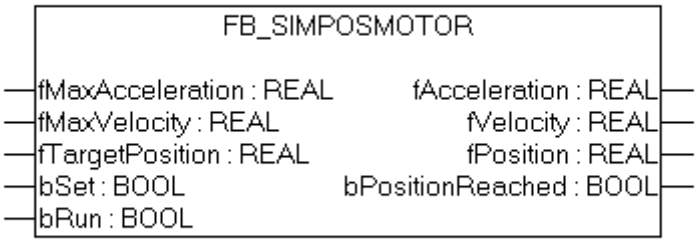
Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib

Development Environment	Target System	PLC Libraries to include
		(Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.8 FB_SimPosMotor

Position-controlled simulation motor.

This simulation motor travels automatically to the entered target position with a ramped velocity profile. The target position 'fTargetPosition' is transferred to 'bSet' with a rising edge. In addition to the present position, the current velocity and acceleration can be accessed at the outputs.



VAR_INPUT

```
VAR_INPUT
    bRun             : BOOL;     (* Motor running *)
    bSet             : BOOL;     (* Acitvates the move command *)
    fTargetPosition  : REAL;     (* Target Position ~(EUnit:mm)*)
    fMaxVelocity     : REAL := 1000;(* Maximal Velocity ~(EUnit:m/s)*)
    fMaxAcceleration : REAL := 100; (* Maximal Acceleration ~(EUnit:mm/s^2*)
END_VAR
```

bRun: Activates the simulation motor

bSet: Adoption of maximum velocity and target position on rising edge

fTargetPosition: Target position in mm

fMaxVelocity: Maximum velocity in mm/s

fMaxAcceleration: Max. acceleration in mm/s^2

VAR_OUTPUT

```
VAR_OUTPUT
    fPosition        : REAL := 0.0; (* Current position ~(EUnit:mm)*)
    bPositionReached : BOOL := TRUE; (* Position reached *)
    fVelocity        : REAL := 0.0;  (* Current velocity ~(EUnit:mm/s)*)
    fAcceleration    : REAL := 0.0;  (* Current acceleration ~(EUnit:mm/s^2)*)
END_VAR
```

fPosition: Current position in mm

bPositionReached: TRUE when the pre-assigned position is reached

fVelocity: Current velocity in mm/s

fAcceleration: Current acceleration in mm/s^2

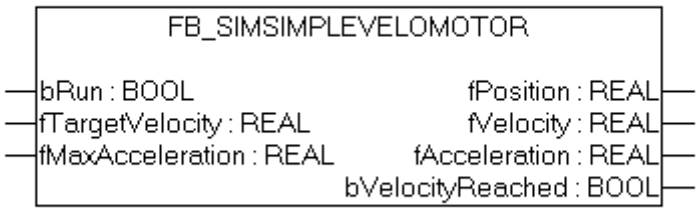
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.9 FB_SimSimpleVeloMotor

Simple velocity-controlled simulation motor.

The target velocity and the maximum acceleration are predefined at the inputs '**fTargetVelocity**' and '**fMaxAcceleration**'. The available outputs are current position '**fPosition**' (as integral of the velocity), current velocity '**fVelocity**' and current acceleration '**fAcceleration**'.



,

VAR_INPUT

```
VAR_INPUT
    bRun              : BOOL;     (* Activates the FB *)
    fTargetVelocity   : REAL;     (* Target Velocity ~(EUnit:mm/s) *)
    fMaxAcceleration  : REAL := 100;  (* Maximal valid acceleration ~(EUnit:mm/s^2 *)
END_VAR
```

bRun: Activates the simulation motor

fTargetVelocity: Target velocity in mm/s

fMaxAcceleration: Max. acceleration in mm/s^2

VAR_OUTPUT

```
VAR_OUTPUT
    fPosition         : REAL := 0.0;     (* Current Position ~(EUnit:mm)*)
    fVelocity         : REAL := 0.0;     (* Current Velocity ~(EUnit:mm/s)*)
    fAcceleration     : REAL := 0.0;     (* Current Acceleration ~(EUnit:mm/s2) *)
    bVelocityReached  : BOOL := TRUE;    (* Velocity is reached*)
END_VAR
```

fPosition: Current position in mm

fVelocity: Current velocity in mm/s

fAcceleration: Current acceleration in mm/s^2

bVelocityReached: TRUE = target velocity reached

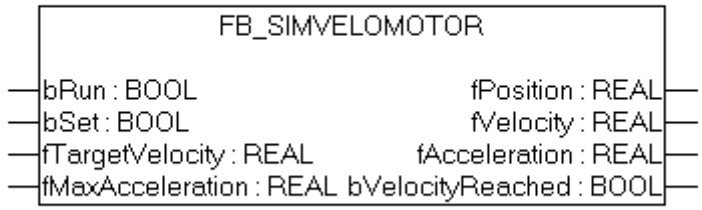
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

6.10 FB_SimVeloMotor

Velocity-controlled simulation motor.

A rising edge in the input '**bSet**' sets the target velocity '**fTargetVelocity**' and the maximum acceleration '**fMaxAcceleration**'. The available outputs are the current position '**fPosition**', the current velocity '**fVelocity**' and the current acceleration '**fAcceleration**'.



VAR_INPUT

```
VAR_INPUT
    bRun             : BOOL;       (* Activates the FB *)
    bSet             : BOOL;       (* Sets the Preset Velcity and Maximum Acceleration *)
    fTargetVelocity  : REAL;       (* Preset Velocity ~(EUnit:mm/s)*)
    fMaxAcceleration : REAL := 100; (* Maximum Acceleration ~(EUnit:mm/s^2*)
END_VAR
```

bRun: Activates the simulation motor

bSet: A rising edge sets the target velocity and the maximum acceleration

fTargetVelocity: Target velocity in mm/s

fMaxAcceleration: Max. acceleration in mm/s^2

VAR_OUTPUT

```
VAR_OUTPUT
    fPosition        : REAL := 0.0; (* Current Position ~(EUnit:mm)*)
    fVelocity        : REAL := 0.0; (* Current Velocity ~(EUnit:mm/s))
    fAcceleration    : REAL := 0.0; (* Current Acceleration ~(EUnit:mm/s^2)*)
    bVelocityReached : BOOL := TRUE; (* Preset velocity is reached *)
END_VAR
```

fPosition: Current position in mm

fVelocity: Current velocity in mm/s

fAcceleration: Current acceleration in mm/s^2

bVelocityReached: Target velocity reached

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.9.0 Build > 1020	PC (i386)	TcSimManager.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

7 First steps

Prerequisites:

- The Original (to be simulated) PLCs projects are ready and the system manager configuration has been created and is activated on the target.
- The PLC projects used should contain the TcSimMan.lib for simulation purposes.
- At minimum one PLC must be free on the local system (local simulation) or on a remote system (distributed simulation).
- All used target systems (local or remote) must have a valid configuration activated.
- The PLC runtimes must contain the Symbol description files
- PLC runtimes must be configured as boot project

see also Requirements for Simulation configurations [▶ 17].

Creation of new TwinCAT Simulation Manager project

Select "File --> New" in the main menu (see Picture 1 [▶ 67])

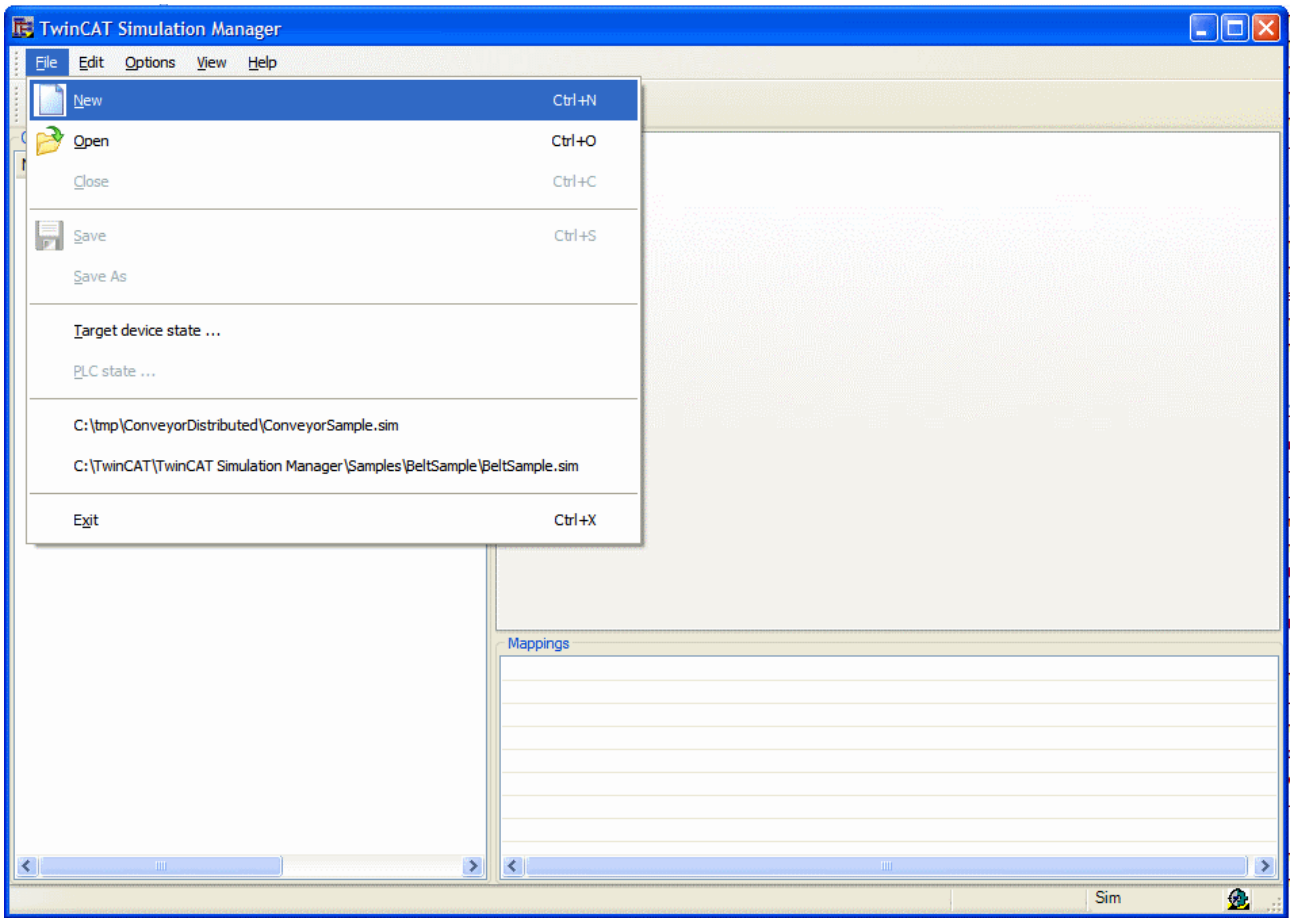


Fig. 52: Picture 1: Open a new project

An "Open new Project" dialog appears and a valid Project name and Location path must be inserted. Press "OK" (see Picture 2 [▶ 67]).

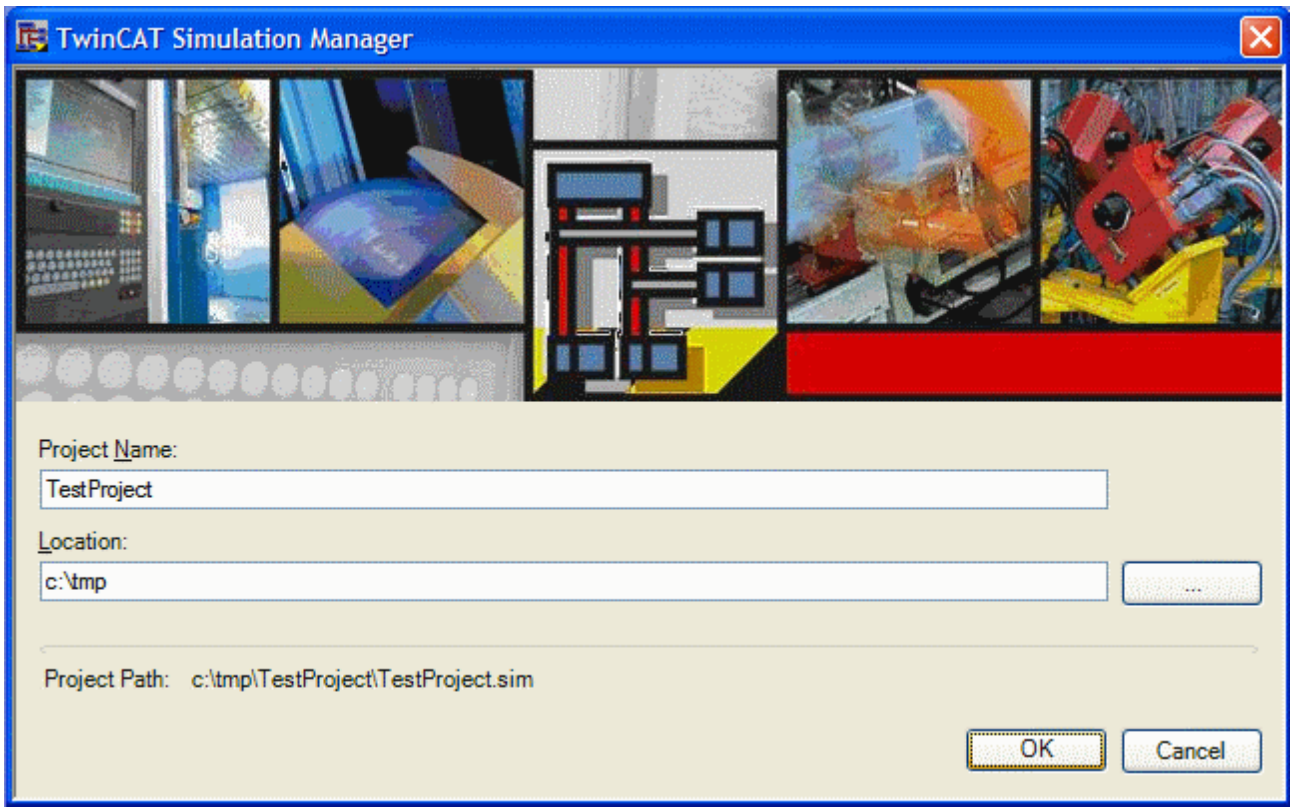


Fig. 53: Picture 2: "Open new project" Dialog

After selecting "OK" the TwinCAT Simulation Manager creates an project folder and project file.

To create a simulation environment the next necessary step will be to select the target systems where the original (to be simulated) projects and configurations exist. For that please activate "Edit --> Select target Systems ..." in the main menu (see Picture 3 [▶ 67]). As reaction the "Add original Systems" Dialog opens (Picture 4 [▶ 67]). The dialog shows all TwinCAT target systems that are found within the connected subnetworks.

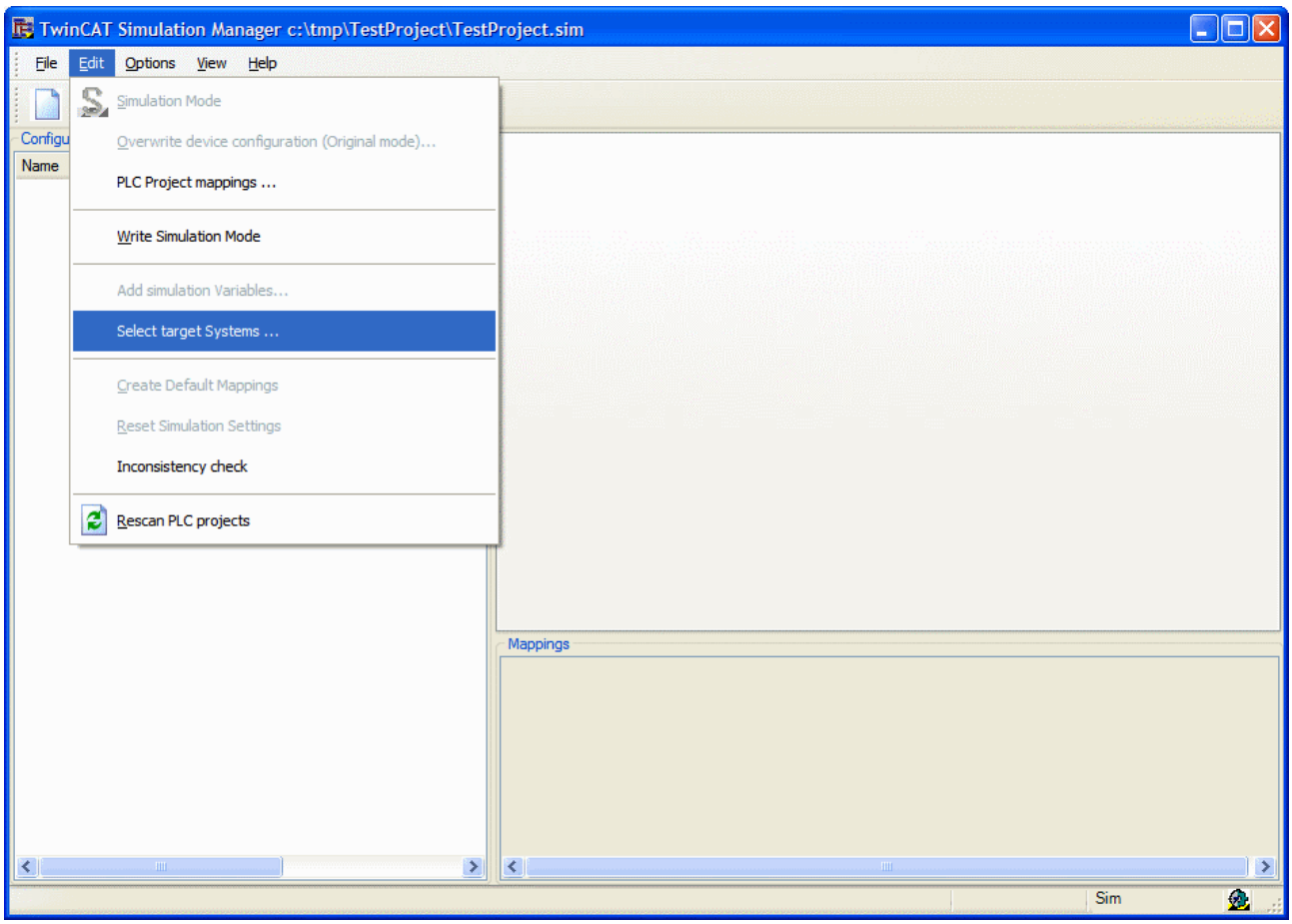


Fig. 54: Picture 3: Open the select target systems dialog

Within this dialog select all systems that are intended to be simulated and add it to the right-hand-side List (Push-Button ">>"). With pressing "OK", the TwinCAT Simulation Manager adds these systems to its own simulation project configuration. If any target system is not assigned to the actual routes of the local system, the TwinCAT simulation manager asks to add this route. If the access to the target systems is possible the actual configuration will be downloaded and cached locally. The same happens for all PLC symbol information (*.tsm files) that are active on the targets.

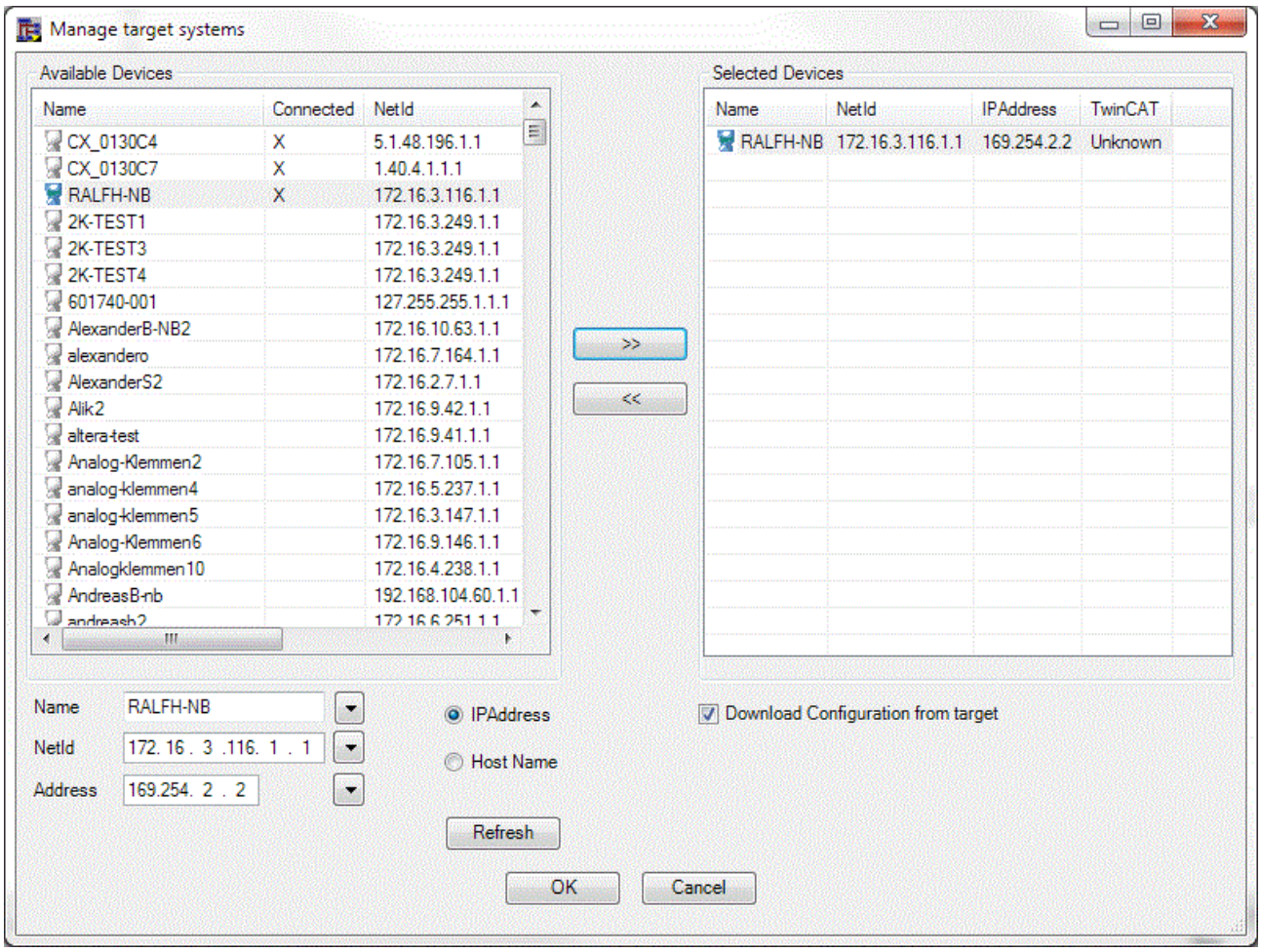


Fig. 55: Picture 4: Selection of the target systems

If all necessary information is found and the data is downloaded successfully the TwinCAT Simulation Manager shows all selected systems within its configuration navigation tree (Picture 5 [▶ 70]).

Creating a new simulation PLC

Up to now, the TwinCAT Simulation Manager only contains the original systems - the ones that must be simulated. Creating and adding a new simulation PLC is done as follows:

1. Select the PLC runtime that is intended to be simulated within the Configuration navigation tree (see Picture 5 [▶ 70])
2. Activate "Create Simulation PLC-Project" from the context menu.
3. Within the now appearing Dialog, the Simulation PLC runtime must be selected (Picture 6 [▶ 70]). This dialog shows the currently available target systems. If the designated simulation system is not in the tree, it can be added via the "Search" button.
4. With leaving the dialog over "OK" the TwinCAT Simulation Manager generates a simulation framework for the selected PLC, assigns it to the specified target system and runtime and adds it to the TwinCAT Simulation Manager configuration project.
   The PLC framework project will contain all located IO-Symbols (see Picture 7 [▶ 70]) of the original PLC project in reversed/mirrored allocation (inputs --> outputs and outputs --> inputs) and is ready to fill with simulation code. The next step is to fill the empty shell of the simulation project with simulation code. This is specific to the application. For this task PLC Control Application can be opened via context menu "Open PlcControl ...".

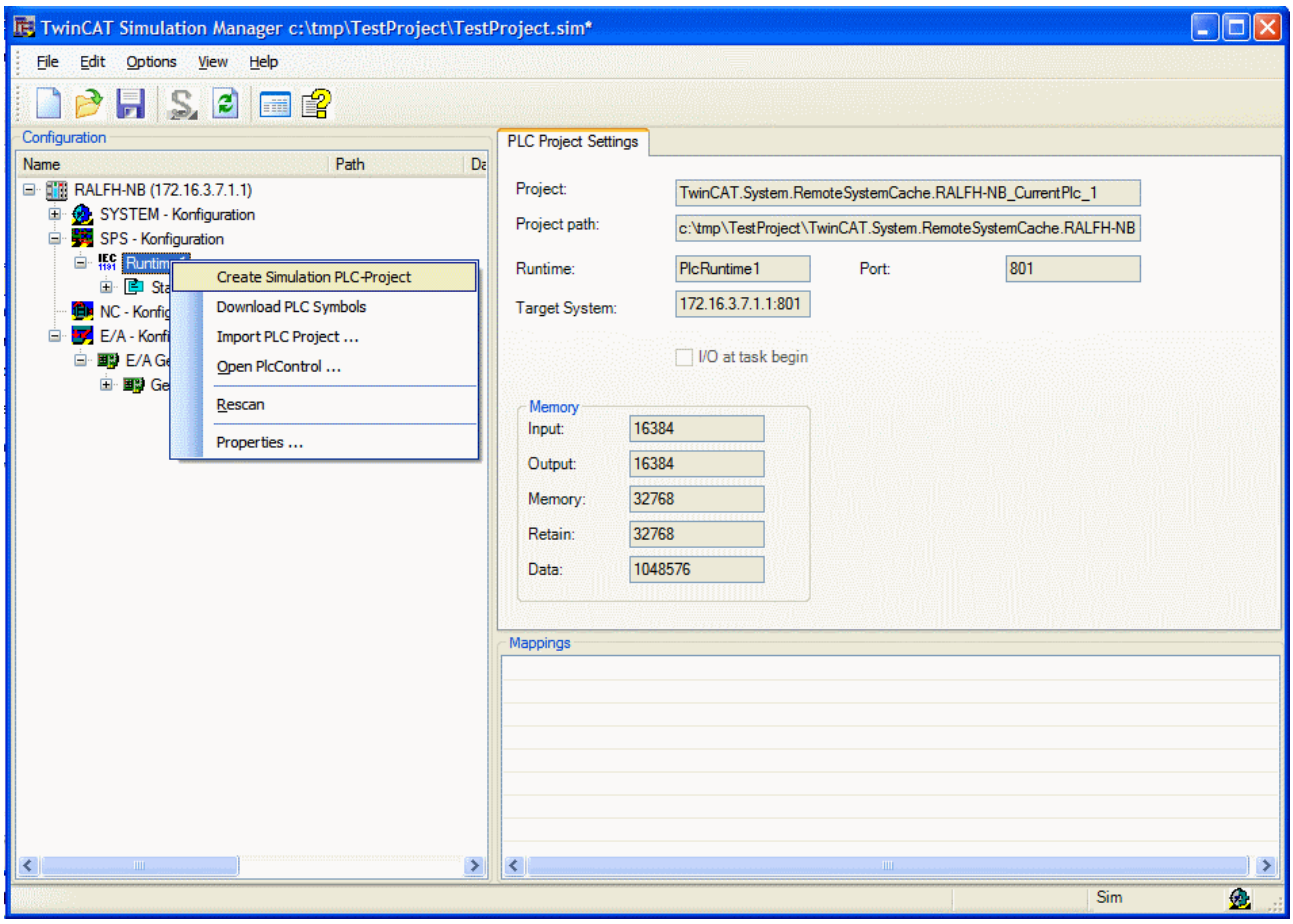


Fig. 56: Picture 5: Create the simulation PLC project

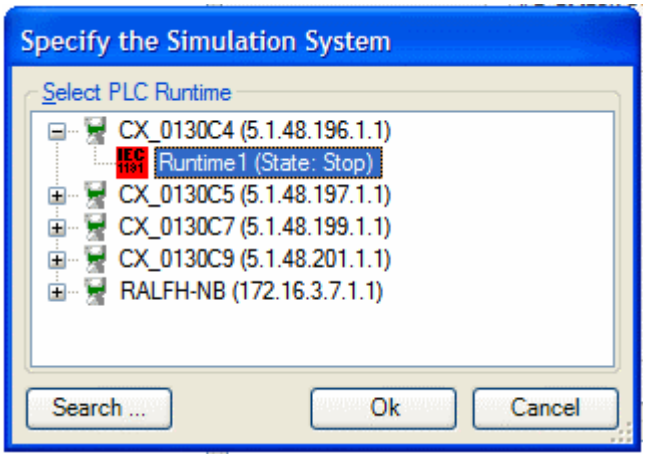


Fig. 57: Picture 6: Selection of the PLC runtime for the simulation

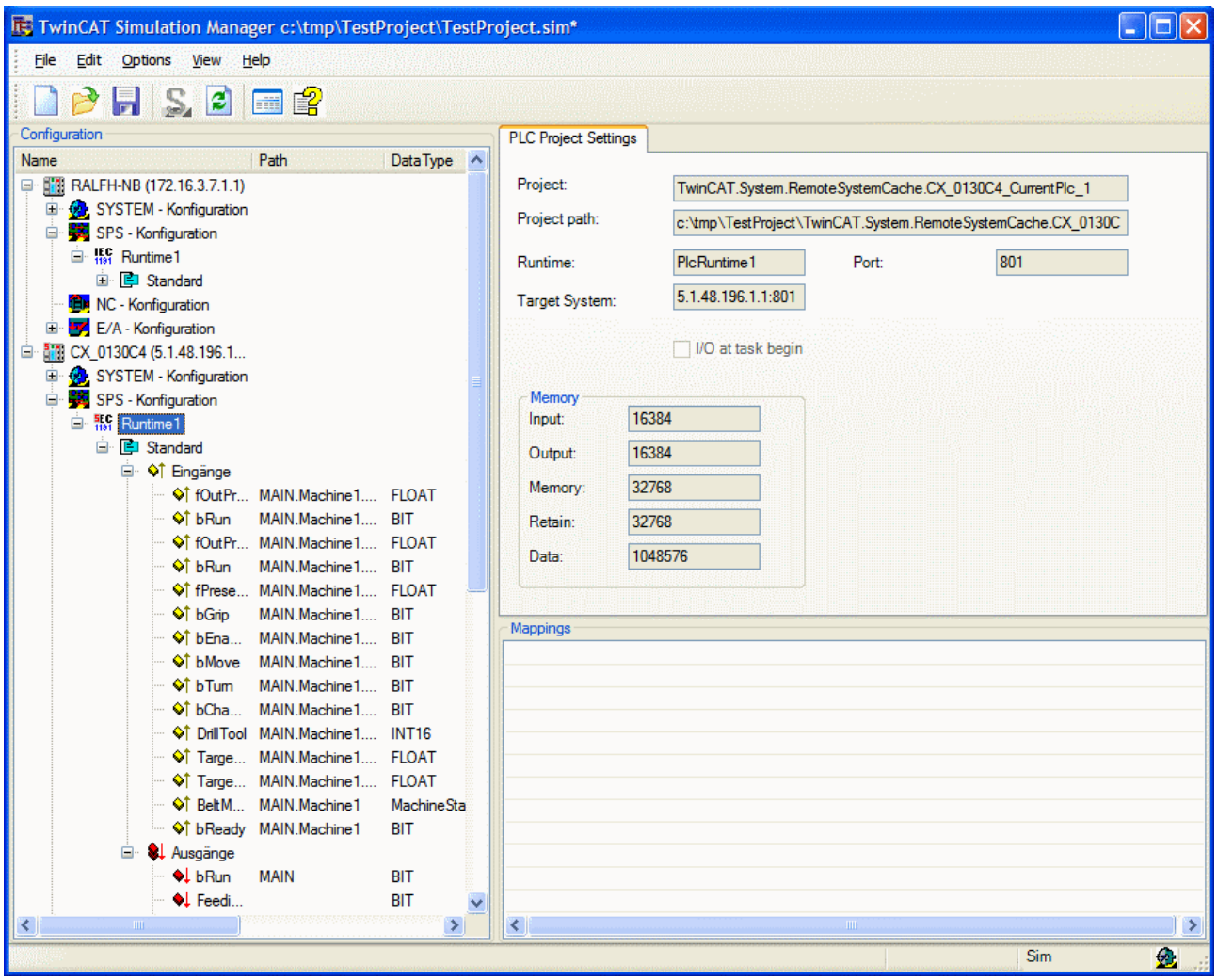


Fig. 58: Picture 7: Created symbol framework for the simulation project

As the Picture 7 [▶ 70] shows, the TwinCAT Simulation Manager code generator generates synonymous symbols with the same name originating from the original project within the simulation project. The TwinCAT Simulation Manager will take advantage of this during computing the default mappings (connecting Original to Simulation symbols) later on.

Inserting an existing Simulation PLC Project

Inserting an existing simulation project can be done as follows.

1. Adding the target system to the list of project systems with calling "Edit -> Select target systems" within the main menu (Picture 3 [▶ 67] and Picture 4 [▶ 67]).
2. Calling "Edit->PLC project mappings" within the main menu (Picture 8 [▶ 72])
3. Assignment of the Simulation project to it's appropriate original PLC project (Picture 9 [▶ 72]).

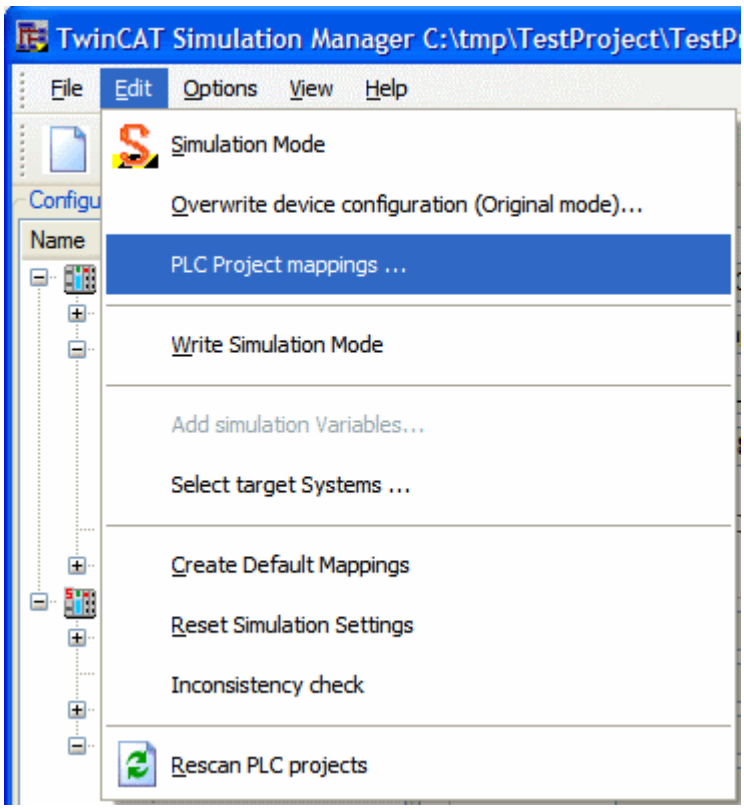


Fig. 59: Picture 8: Activating the "Plc Project Mappings" Dialog

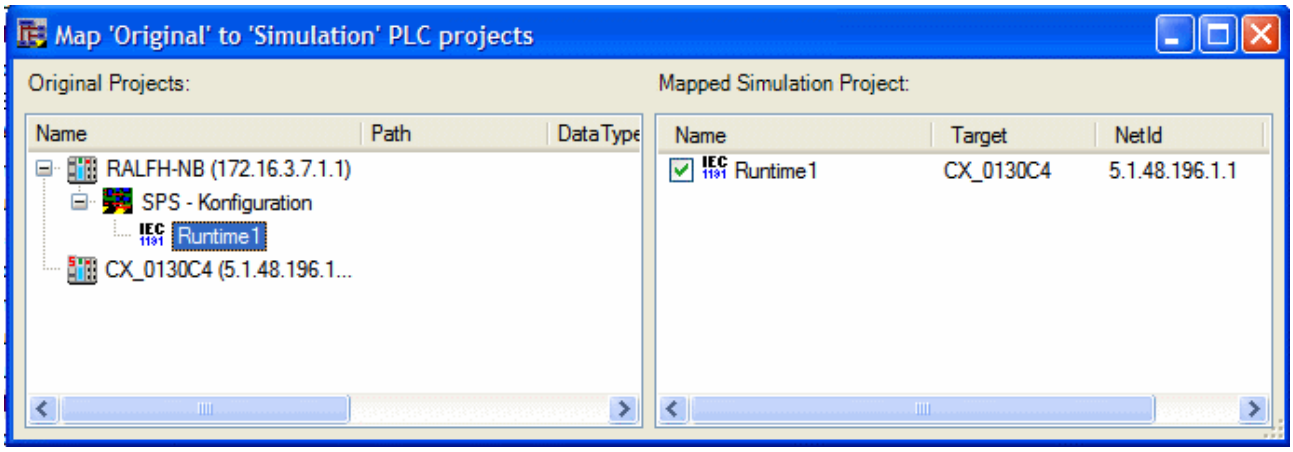


Fig. 60: Picture 9: Mapping the Original PIC to the Simulation PLC

Configuration of the simulation mode

The project state is now as follows:

The original and simulation target systems are designated and at least the interfaces of the simulation functionality generated. For supporting not existing hardware, the simulation settings must be adjusted. That means:

- Disabling of nonexistent Devices
- Simulation of non-existing Axes and Drives
- Replacing of IO functionality by the simulation PLC

The devices will be disabled like shown in Picture 10 [▶ 73] and Picture 11 [▶ 73]. The red 'S' marks the hardware that is deactivated in simulation mode.

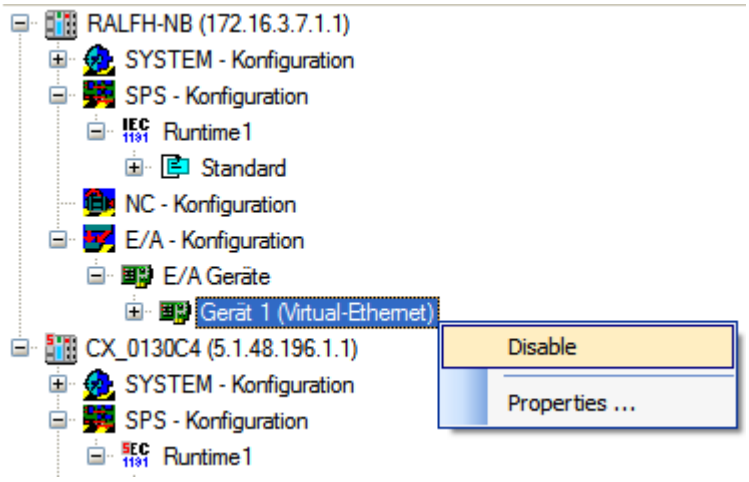


Fig. 61: Picture 10: Deactivation of the nonexistent hardware

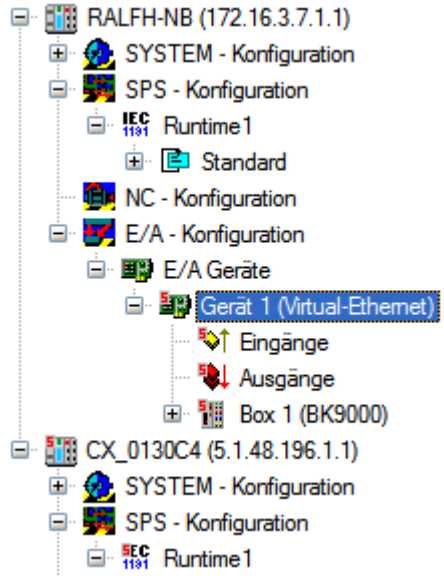


Fig. 62: Picture 11: Disabled hardware

The simulation of Axes will be configured via the Context Menu within the Axis list view (Picture 12 [▶ 73]).

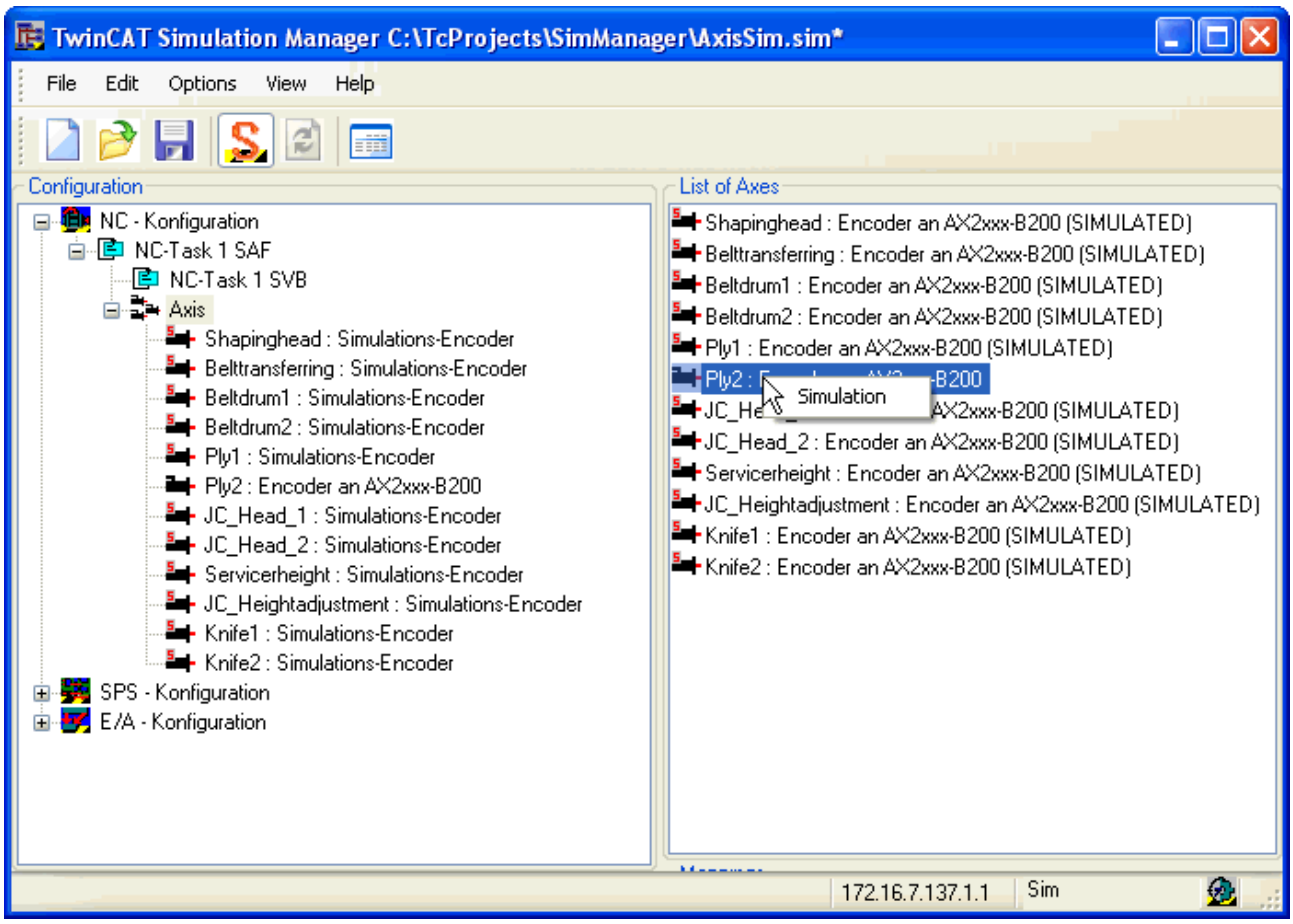


Fig. 63: Picture 12: Simulation of Axes and Drives

Finally, the created or added simulation PLC projects must be bound to the orginal PLCs. There are two options to do this task:

1. Manually adjustment Symbol by Symbol or
2. Using a default mapping algorithm

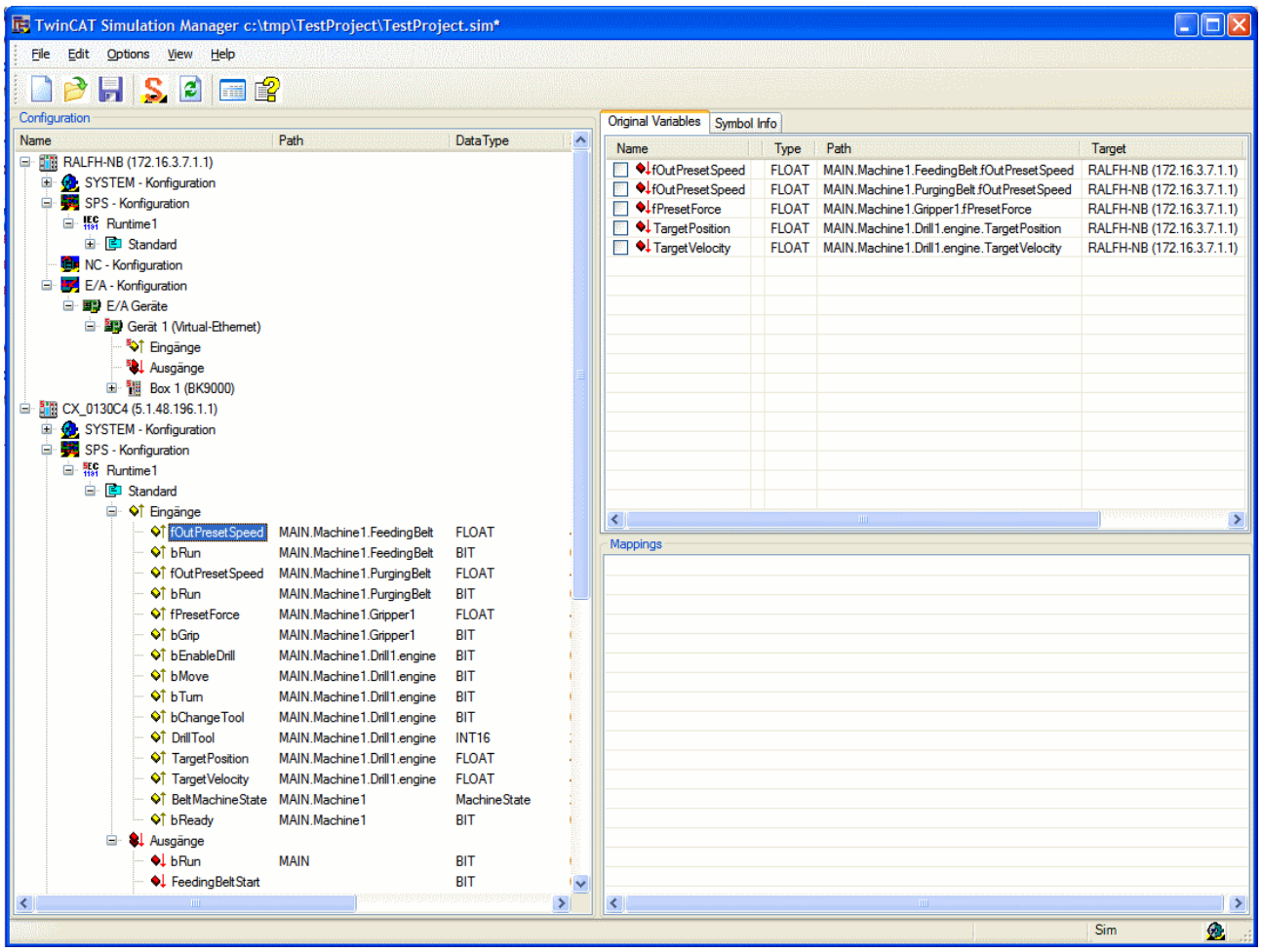


Fig. 64: Picture 13: Simlation Symbols without mapping to the original PLC

The manual configuration works simply be clicking the checkboxes (or context menu) on the appropriate symbols (see Picture 13 [▶ 73]) within the mapping targets view (upper right-hand-side list box). Dependent of the selection of the symbol in the left-hand-side "Configuration" navigation tree the mapping targets view collects all "Original" or "Simulation" symbols with the same datatype within its list. Potentially every single symbol of these target symbols can be bound to the source symbol.

The automatic configuration happens if "Create Default Mappings" is activated from the "Edit" main menu of the TwinCAT Simulation Manager (Picture 14 [▶ 73]). This functionality iterates over all Symbols within the original PLC projects and binds it to simulation symbols of the same symbolic name / access path, same datatype but reversed location (see Picture 15 [▶ 73]). This should be the default option because it ends up with a good standard configuration that can be fine tuned manually afterwards.

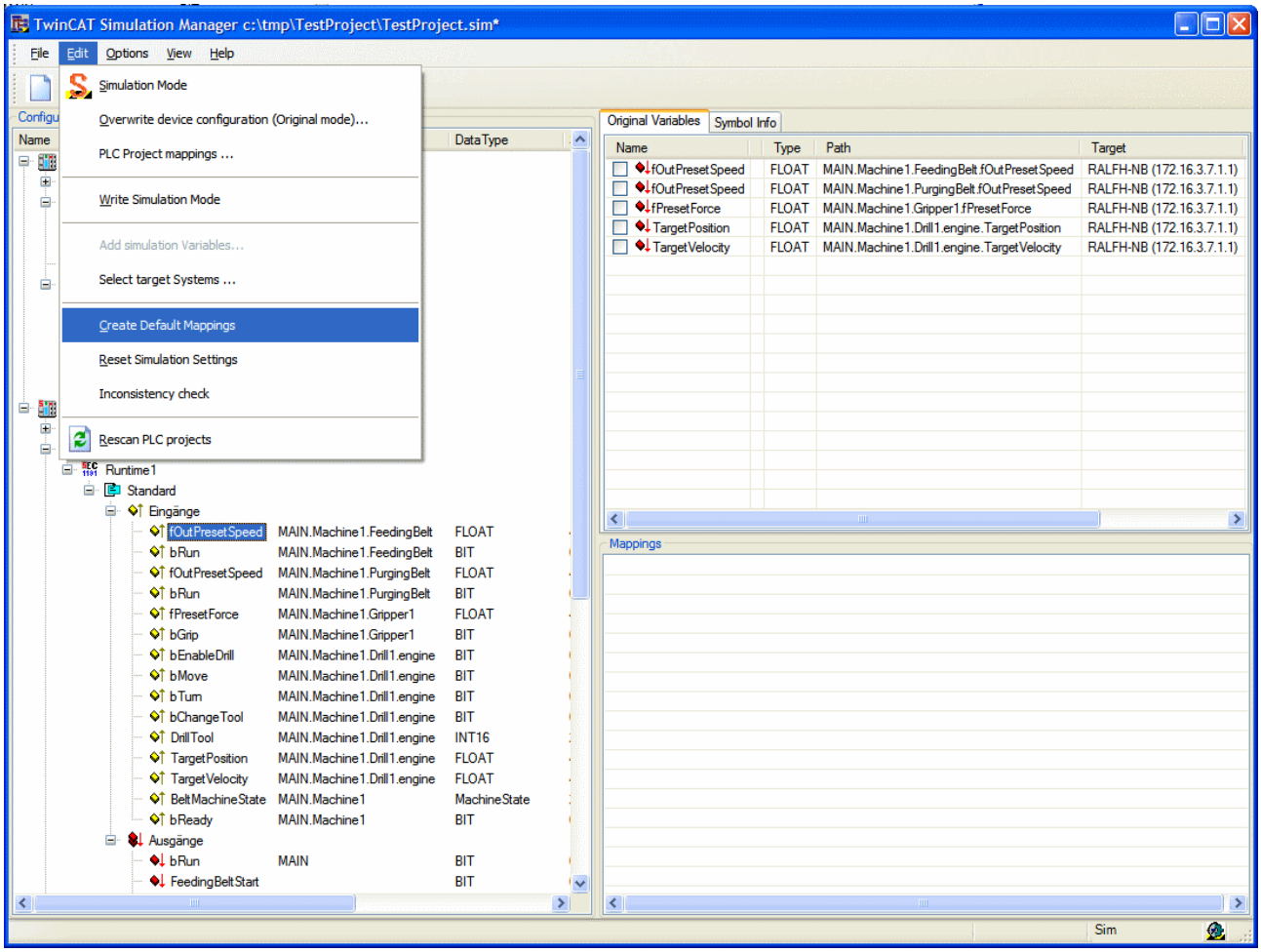


Fig. 65: Picture 14: Automatic creation of default mappings

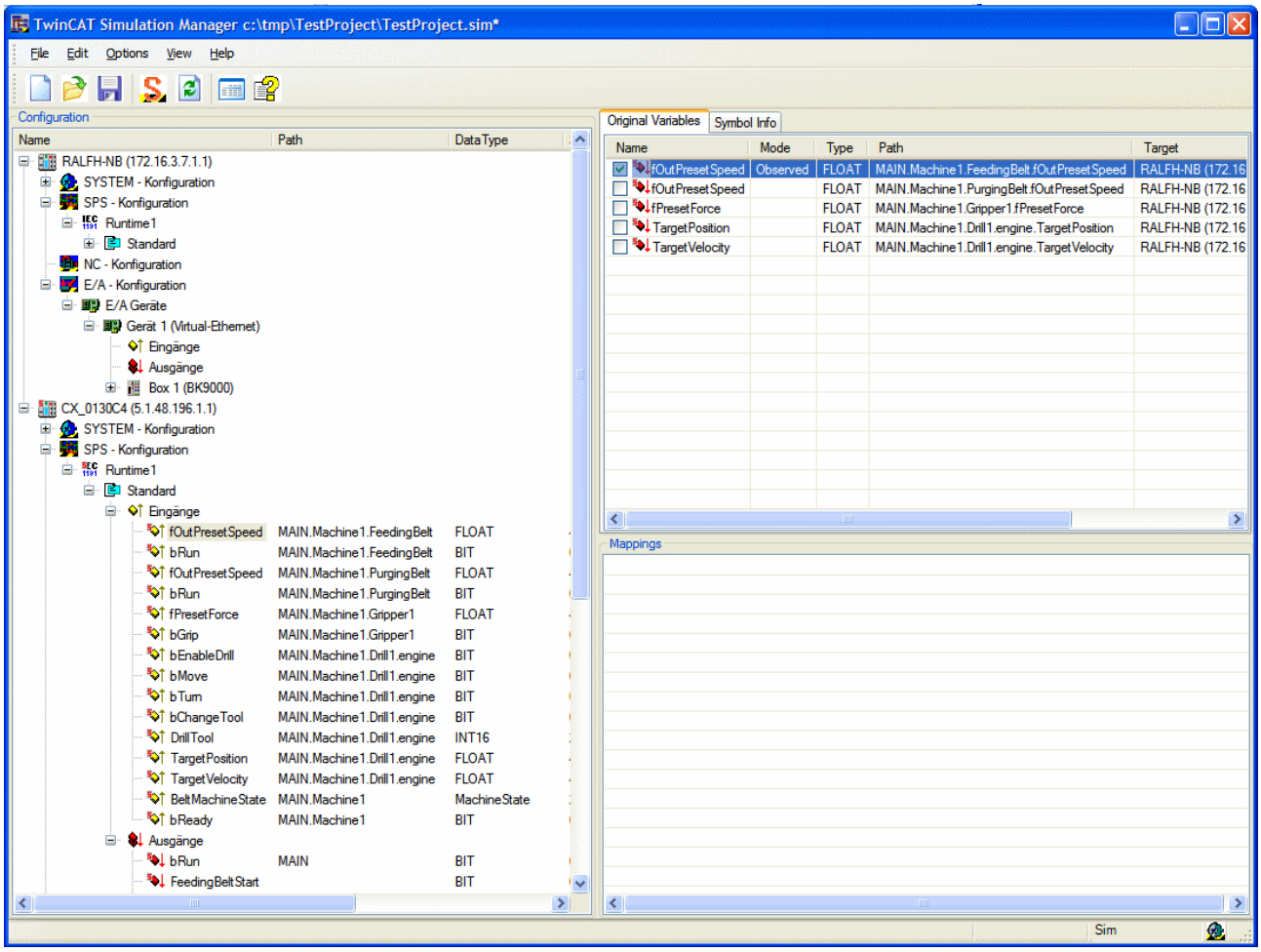


Fig. 66: Picture 15: Simulation mappings set to the original Symbols

Switching the Simlation-Machine-Modes (Original <--> SImulation)

If the configuration of the Simulation project is complete, the the mode of the system can be switched to Simulation Mode and back by simple Activation over the Main Menue "Edit --> SimulationMode" or the Simulation Symbol within the TwinCAT Simulation Manager ControlBar.

Because the switchover process includes several steps that usually need (safety) confirmation(s), the TwinCAT Simulation Manager volunteers the User by an intelligent Machine configuration wizard step by step.

The descriptions of this wizard and the specifics of the machine reconfiguration and restarting can be found here:

- Switching over to simulation mode [▶ 25]
- Switching back to original mode [▶ 28]

Resetting the simulation settings

The Simulation settings can be removed from the main menu "Edit--> Reset Simulation Settings ..."

8 FAQs (frequently asked questions)

The system reports "The system manager configuration file 'XXX.wsm' cannot be opened. The file is probably blocked by another system manager instance!"

Problem:

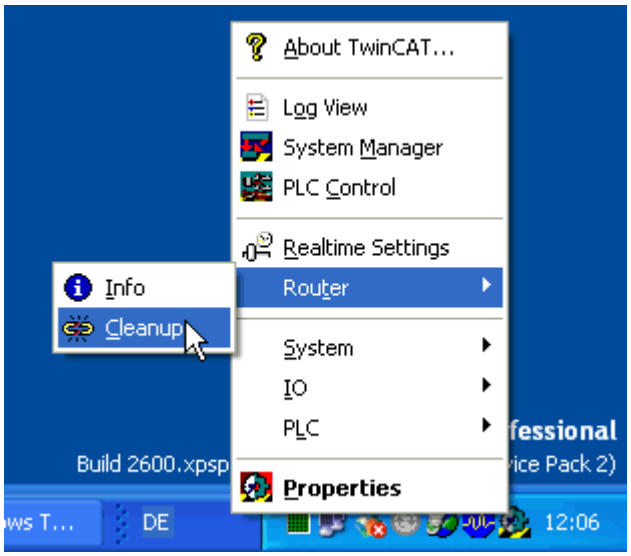
The 'TwinCAT System Manager' is addressed by the 'TwinCAT Simulation Manager' as the 'Out-of-process' COM server. They run as two separate processes (EXE files) that communicate with each other.

This fault message is displayed when a system manager process can no longer be addressed. The system is blocked as a second system manager instance cannot be started on the same configuration file.

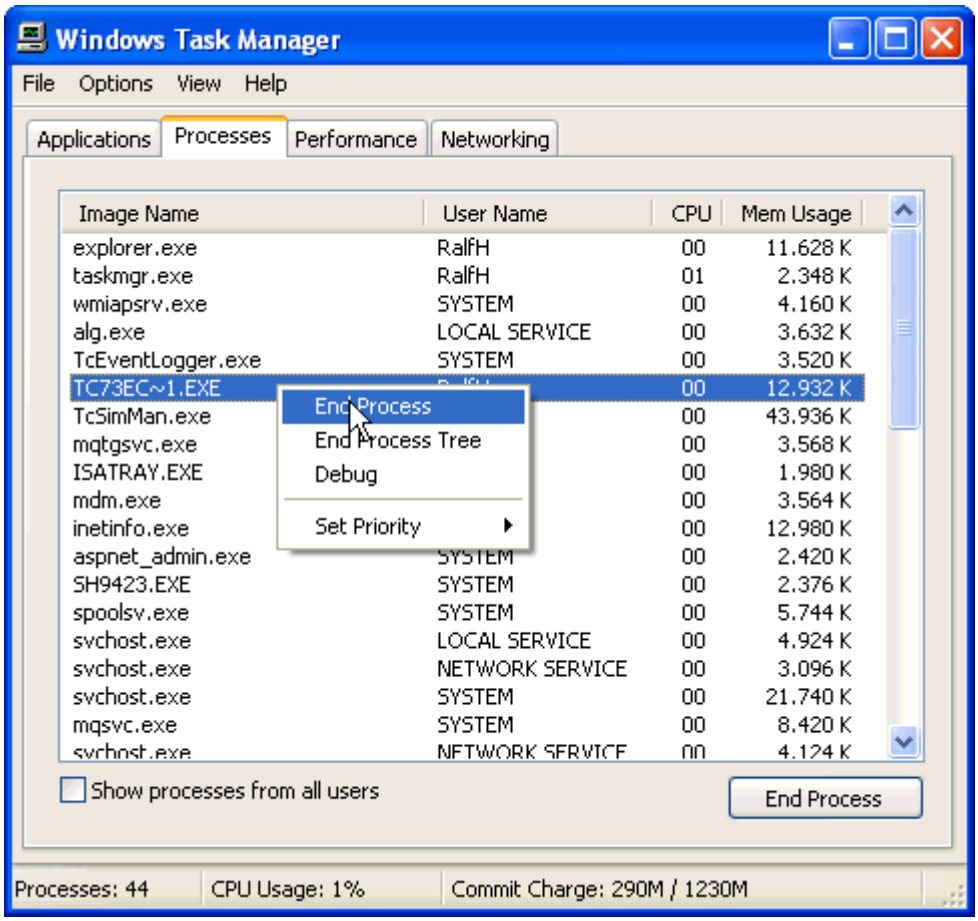
Solution:

The blocked system manager process must be exited. To this end, please carry out the following operations:

1. Router cleanup (in some cases this step alone is enough)



2. Exit the system manager process.



3. If the system manager process does not exit, another router cleanup should be initiated. This should terminate the system manager process!

Error Message:
"The PlcControl symbol file for the project 'XXX' has not been found. The system manager configuration must be corrected before it can be used!"

Problem:

The symbol file (*.tpy) specified in the system manager could not be found!

Solution:

Reconnect the symbol file in the system manager configuration and rescan the symbol ("Rescan").

The newly created simulation PLC project does not contain any inputs or outputs.

Problem:

The following conditions must be fulfilled to create the allocated variables:

- The original PLC projects must contain allocated variables, which can then be exported into the simulation PLC project.
- The simulation PLC symbols must be used in the simulation project
- The simulation PLC project must have been compiled.

Solution:

Please check whether symbols have been created in the simulation PLC and that these can also be used, as the code creator ignores unused symbols during compilation. Calls of program POEs are not created in the simulation PLC, they will possibly have to be retrospectively taken care of. .

Make sure that the used symbol address ranges are big enough. The PLC Control Compiler creates the following error message:

Error: Variable 'X' to large for address %ADDRESS

In this case increase the memory in the project options of the relevant PLC project.

When the simulation PLC project has been compiled without any errors, the symbols in the simulation manager can be updated at the corresponding PLC node with "Re-load".

More Information:
www.beckhoff.com/ts1110

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com