CODESYS V3.5 SP4
Features and improvements

Agenda

1. Runtime
2. Motion+CNC
3. Visualization
4. Engineering
5. Safety
6. Fieldbus

Agenda

Runtime system

Overview

- Delivery Manager
- Device licensing
- Online help for runtime system API reference
- Exception position can be determined purely from the logger

Delivery Manager

- To optimize delivery of the CODESYS Runtime Toolkit for the adaptation to customer devices
- Completely based on Python (platform independent)
- Web frontend for configuration and handling
- Delivery via local PC possible
- Each delivery is completely logged and can be repeated any time.
- The device is entirely described in the device profile.
- All necessary files are created (DevDesc, cfg file, header files, component list etc.), “Build“ can be enabled optionally

Device licensing

- Features / products can be licensed per single device
- Based on WIBU technology:
  - WIBU Key (USB, SD Card, CF Card)
    - Available under Windows, WindowsCE and Linux* (USB connection subject to strong platform restrictions)
  - WIBU SoftKey (“ActLicense“, without WIBU hardware)
    - Based on a clear device identification/code. Currently available: for Windows and Linux based on the WIBU SmartBind mechanism
    - In the future:
      Establishment of a serial device number on all controllers
    - This serial number is necessary for the ActLicense to be used.

Device licensing

- The license can be purchased in the CODESYS Store.
- Options for license activation:
  - Direct connection from CODESYS both to the controller and to the internet
  - Step by step via context file
    1. Get the license information from the controller (connection to the controller)
    2. Download the license file (connection to the internet)
    3. Import the license file (connection to the controller)
  - Direct import of the license into the USB WIBU Key on the PC
- The license protection via CODESYS License Manager is no longer valid
  ➔ Version update requires updated runtime licenses,
  e.g. for CODESYS Control RTE

Online help for runtime system API reference

- Replaces unhandy document CODESYSControlV3_Reference.pdf
- In the future:
  - Central help system for the runtime system (reference, feature description, manual)

Exception position can be determined purely from the logger

- Exception position is saved in the logger:
  "*SOURCEPOSITION* App=[<app>%s</app>] area=<area>%d</area>,
  offset=<off>%ld</off>“
- Helps localizing and finding crashes
- Current version of the runtime system necessary

Agenda

1. Runtime
2. Motion+CNC
3. Visualization
4. Engineering
5. Safety
6. Fieldbus

Overview

- Improvements
- New drive drivers
- New transformations
- SoftMotion Light

Improvements

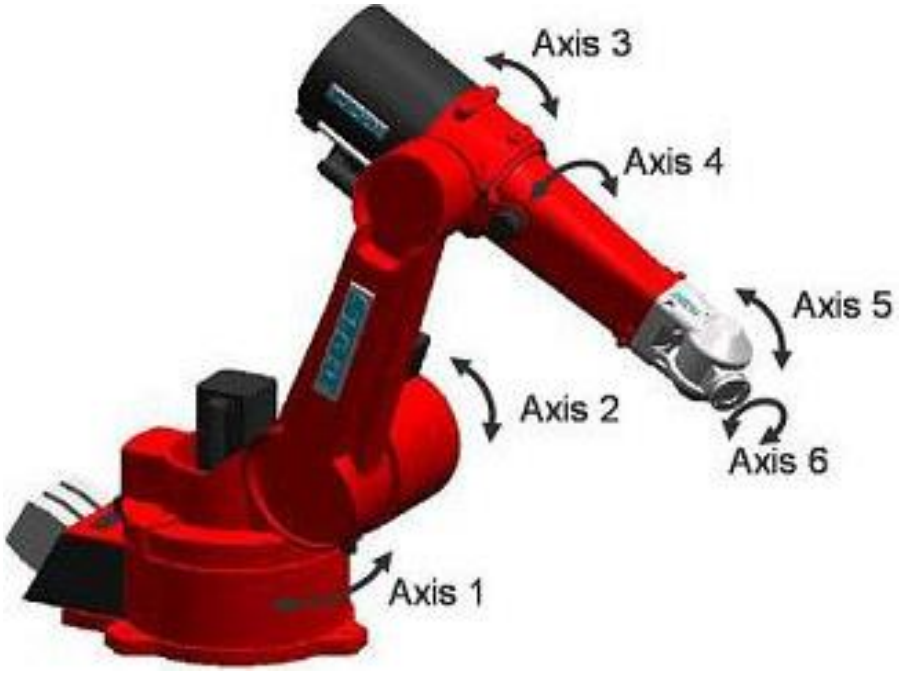
- X-Interpolator with 3D path
- GearInPos: avoid reversing
- Support of a 32 bit overflow of the position for finite axes

Drive drivers

- Kollmorgen AKD (EtherCAT)
- CMZ SD (CAN)
- Festo CMMP (CAN)
- *Servotronix CDHD (EtherCAT)* – in preparation

Motion+CNC

- 4-axis palletizing robot (4 rotatory axes, the gripper is aligned mechanically and horizontally
- 6-axis articulated robot with a central gripper (6 rotatory axes, the three coordination axes intersect in one point.



CODESYS SoftMotion Light - introduction

- Simple one-axis motions on many axes
- SoftMotion Light: commands / surveys a motion
- In the drive: cyclic preselected target value / trajectory calculation
- Consequence: small bus load, small processing load for the controller

Motion+CNC

	Cyclic preselected target value (CODESYS SoftMotion)	Acyclic setting (CODESYS SoftMotion Light)
Required computing power	high	low
Required fieldbus width//fieldbus speed	high	low
Required realtime property (controller + fieldbus)	high	low
Synchronization of several axes (CNC, cam, gear)	yes	no
Prompt reaction to new commands	high	medium/low (fieldbus/depends on configuration)

CODESYS SoftMotion Light – scope of delivery

- Configuration in CODESYS with extra library, similar to CODESYS SoftMotion
- Single axis movement according to PLCopen:
  - MC_MoveAbsolute/MC_MoveRelative
  - MC_MoveVelocity
  - MC_Stop/MC_Halt
  - MC_Home
- Other POUs
  - MC_Power, MC_Reset, MC_ReadStatus
  - SML_ReinitAxis, SML_ChangeAxisConfig
- Visualization templates as in CODESYS SoftMotion
- Help for commissioning (SML_StartupDrive)
- Project for testing the compatibility of drives

Agenda

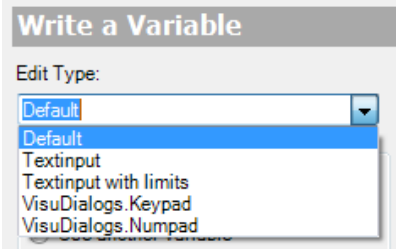
1. Runtime
2. Motion+CNC
3. Visualization
4. Engineering
5. Safety
6. Fieldbus

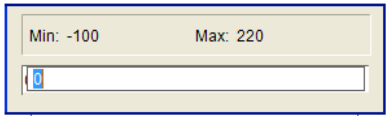
Visualization

Overview

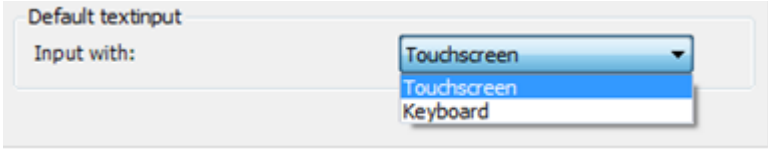
- Optimized value input
- Unit conversion
- Rotation of images and texts
- Usability

Optimized value input



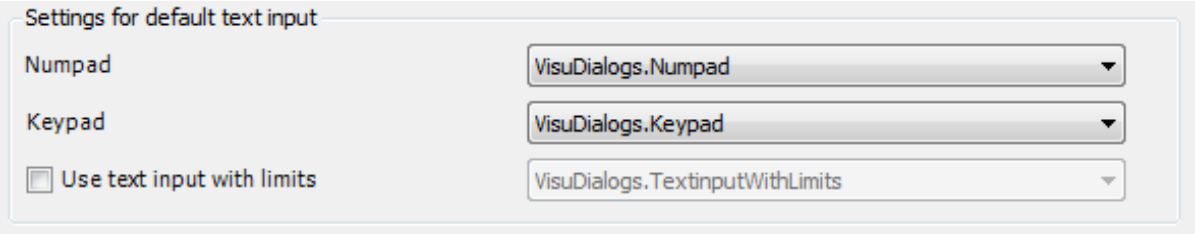


- Two new input types:
  - “Textinput with limits”
  - “Default”
- The input type “Textinput with limits” opens a dialog:
- Dialog
  - is in VisuDialogs
  - can be freely configured
- Input type “Default”: Definition of the input option upon usage in the client



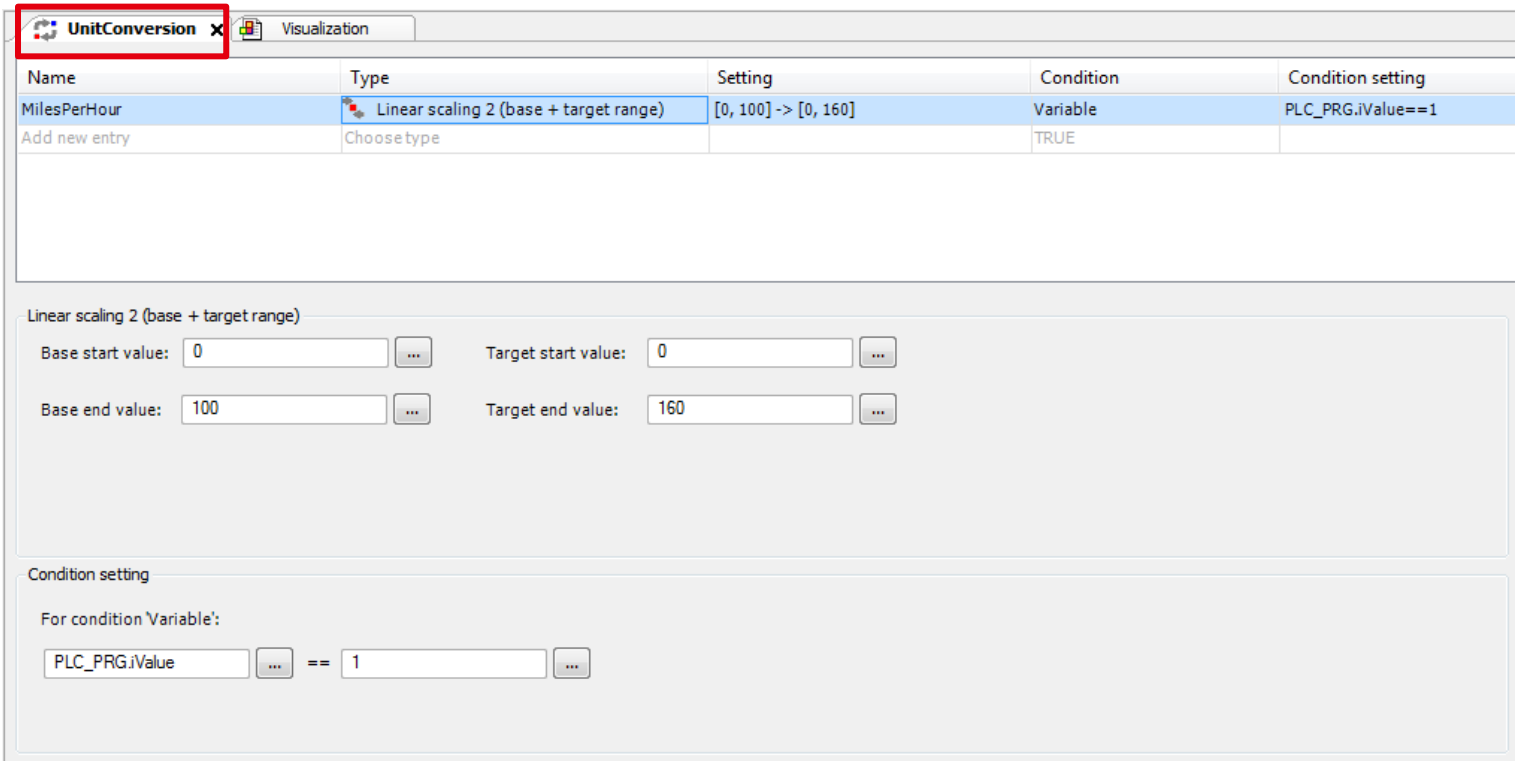
Optimized value input

- Visualization Manager:
  Direct selection of input dialogs under “Default” settings



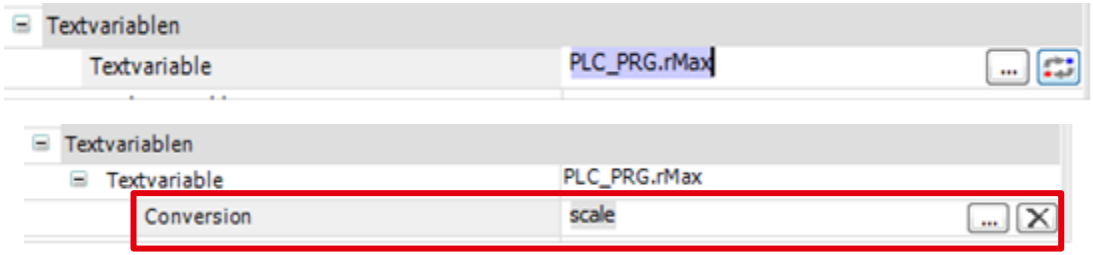
Unit conversion

- New object type: “Unit Conversion”
- Generation of an FB per created conversion
- Automatic generation of a global variable per conversion name



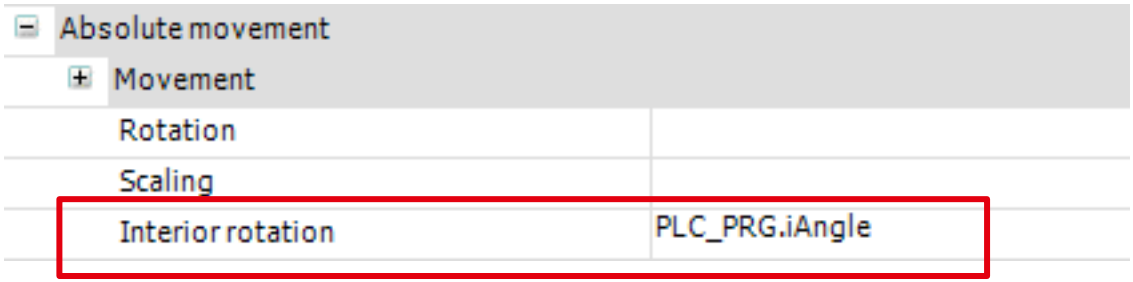
Unit conversion

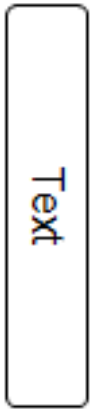
- Usage:
  - Within the application
  - In the visualization
  - Separation of value display from the value content e.g. for different countries



Rotation of images and texts

- New: “Interior rotation” of images and texts

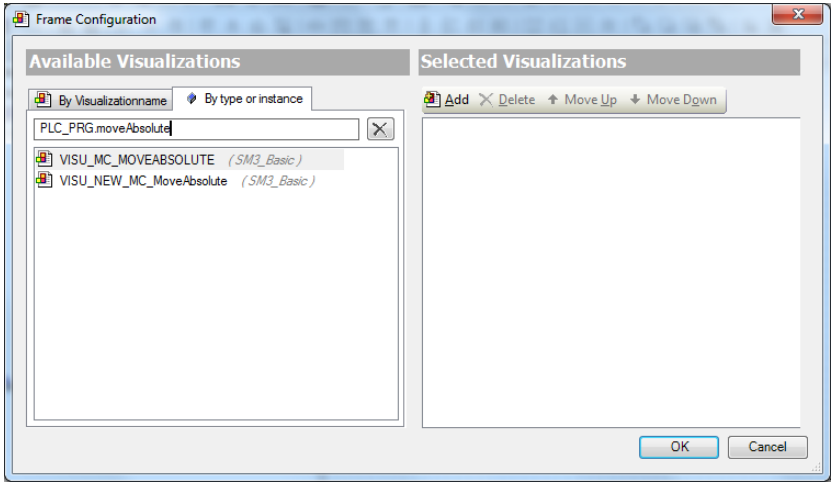


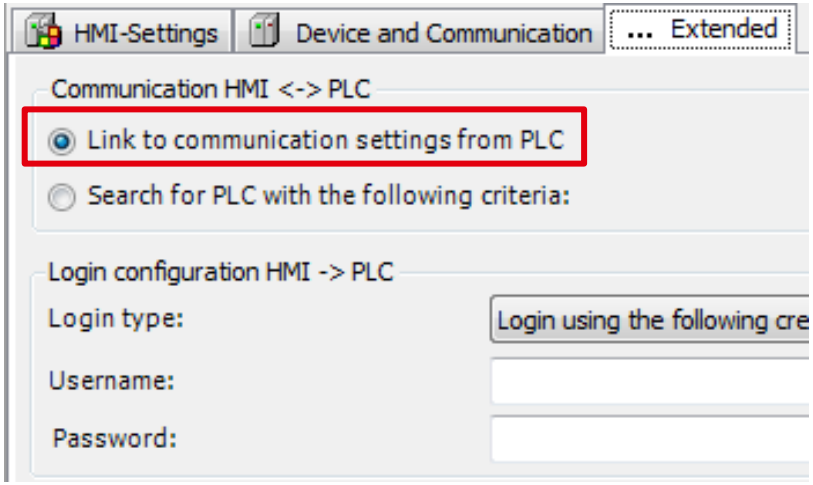


- Usage: Scale labeling, rotation of pointers etc.

Usability

- Simplified usage of the frame element:
  Selection of the referenced Visualization through variable instance





- Simplified communication settings in CODESYS HMI:
  Use of current communication settings
  (device name, IP address)

Agenda

1. Runtime
2. Motion+CNC
3. Visualization
4. Engineering
5. Safety
6. Fieldbus

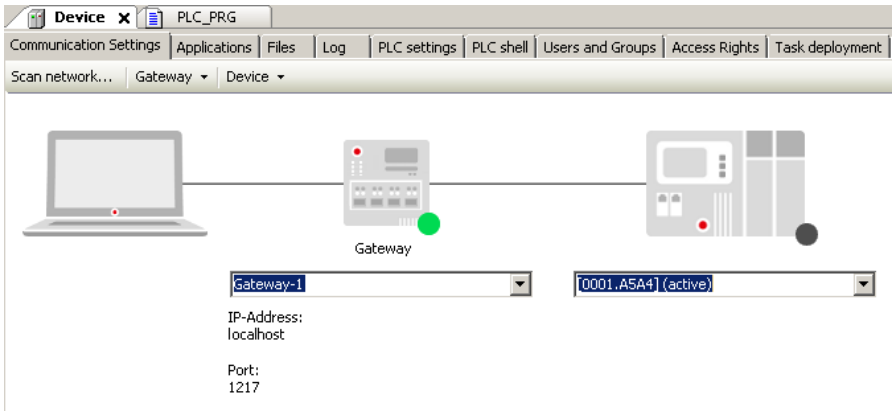
Engineering

Overview

- Usability improvements
- Improved library concept
- Performance, compiler, debugging and language model improvements
- Improved CODESYS Application Composer

Usability improvements

- CFC editor:
  - Grouping of elements
  - Support of flow control
- Communication dialog:
  - Significantly simplified
  - Saving devices or their IP address
  - Selection of favorite devices per drop-down



Libraries – proven concepts

- Comprehensive guidelines for library creation
- Place holder concept for implementation libraries
  - Instead of referencing a specific version, insertion of a place holder
  - Place holder resolved by device or library profile (usually depends on compiler version)
  - Ensures only one version of a certain library is used.
- Modular concept for interface libraries
  - Data types and interfaces always used in latest version
  - And thus suitable for all other libraries
  - Prerequisite: Only compatible extensions (check available in CODESYS)

Libraries – problems

- Place holders not suitable for end users
  - No access to the device description
  - No access to the library profile
- Place holders not really suitable for library environment in the CODESYS Store
  - Access to the library profile possible but dependent on the compiler version
- GUI not really helpful for avoiding errors

Libraries – solution

- Introduction of “free“ place holders
  - No resolution through device description or library profile
  - Selection of version in library manager by user
  - Updates offered in the “Project environment“ dialog
- Library documentation in library source code
  - Implementation libraries (→ should be inserted using place holders)
  - Interface libraries (→ should be inserted using “asterisk”)
  - Container libraries (→ should be inserted using a specific version)
- Adding library references significantly simplified

Performance improvements

- Fast online change upon
  - code change
  - change of initial values (not constants!)
  - new local variables in functions, methods and programs
  - new functionality
- Boot project after online change

Expert debugging

- Debugging crashes
  - New: Details in the dialog
  - New: Crash position in logger with GOTO

Conditional breakpoints

- Conditional breakpoints
  - Any Boolean expression (e.g: a[i] > p^.component)
  - In addition to the conditions available until now
  - Hit counter only counts if condition is correct

- Requirements
  - CODESYS 3.5.4.0
  - CODESYS Control 3.5.4.0 (version in device description)
  - Monitoring2 component

Systemoperator for tick count

- __GETLTICK generates system tick
- Performance advantage compared to SysGetTime
- Implemented for x86, x86-64bit, ARM, Cortex, PPC and MIPS
- Call of SysGetTime, if not supported
- Requirement: Library for recognition of clocking frequency

Short circuiting

- Expressions with AND / OR: execution/calculation of both operands
  Example: pTest <> NULL AND pTest^.bTest
- Due to compatibility reasons the logics of this expression must remain unchanged.
- New operators:
  - “AND_THEN“
  - “OR_ELSE“
  - Execution of the second operand only upon logical FALSE of the first operand
- Possible performance advantages

Compilers

- Additional parameters for derived FB_Init
- PowerPC: Support of e500v2 core (double precision vector unit)
- ARM/x86: C compatible call interface (not used by CODESYS itself)
- ARM: unaligned access with memcopy

CODESYS Application Composer – improvements

- New license model: workstation licenses
- Search in module instances
- Persistence Manager: shorter loading and saving times
- Sequence editor: Display of any desired FB variables in online mode

CODESYS Application Composer – alarm generator

- Modules: Definition of alarms
- Generator: Creation of the CODESYS alarm configuration from definitions
- Modules: Individual alarm reaction
- Generator: Creation of visualizations with alarm table

Agenda

Safety

Overview

- CODESYS Safety (for SIL3 safety controllers)
- CODESYS Safety SIL2

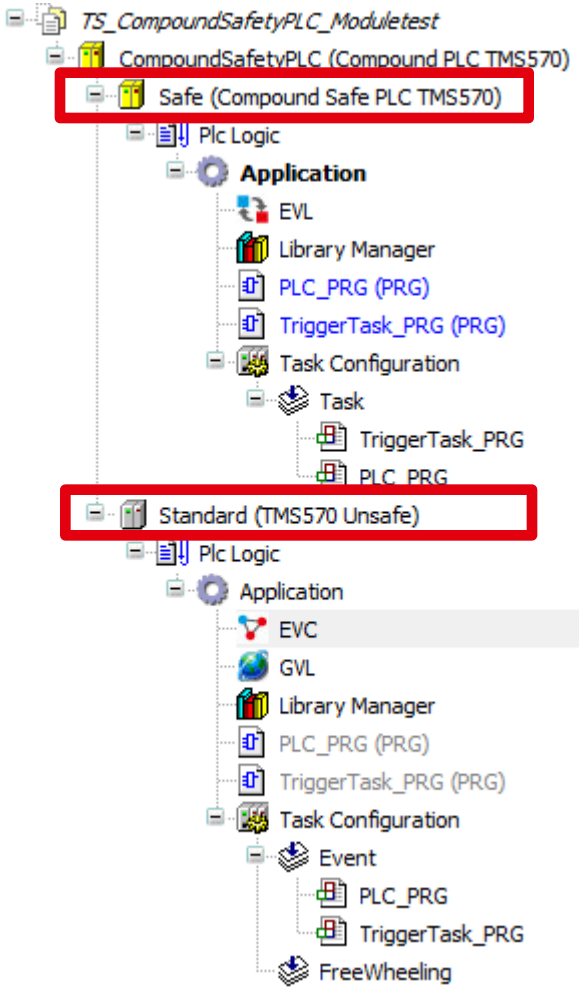
CODESYS Safety (for SIL3 safety controllers)

- Version cycle decoupled from Service Pack ➔ Release in January
- Release FSoE protocol stack
- Release GSDML converter for Profinet
- All releases on CODESYS V3.5 SP4

Safety

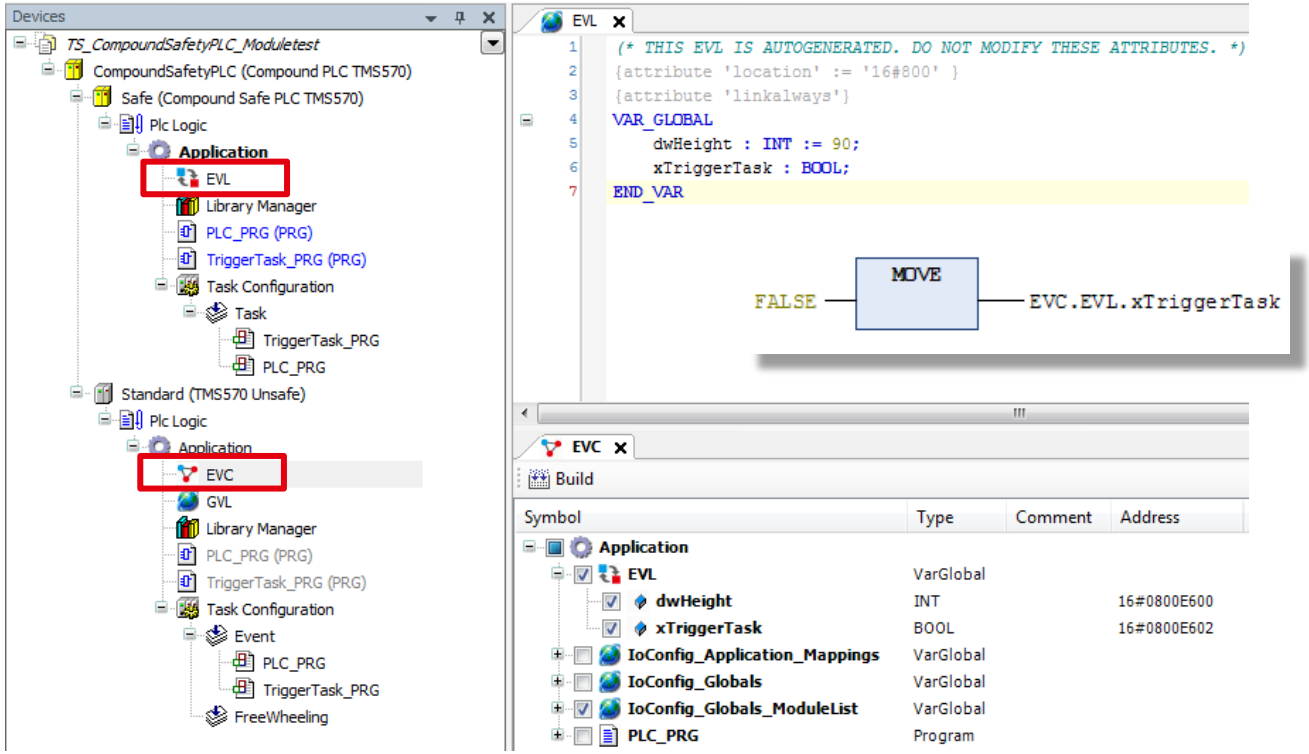
CODESYS Safety SIL2

- Combination of safe and unsafe applications
  - Separation of unsafe code in second application Separation of unsafe I/O configuration in second logical device
  - Possibility to run both runtimes on one or two CPUs



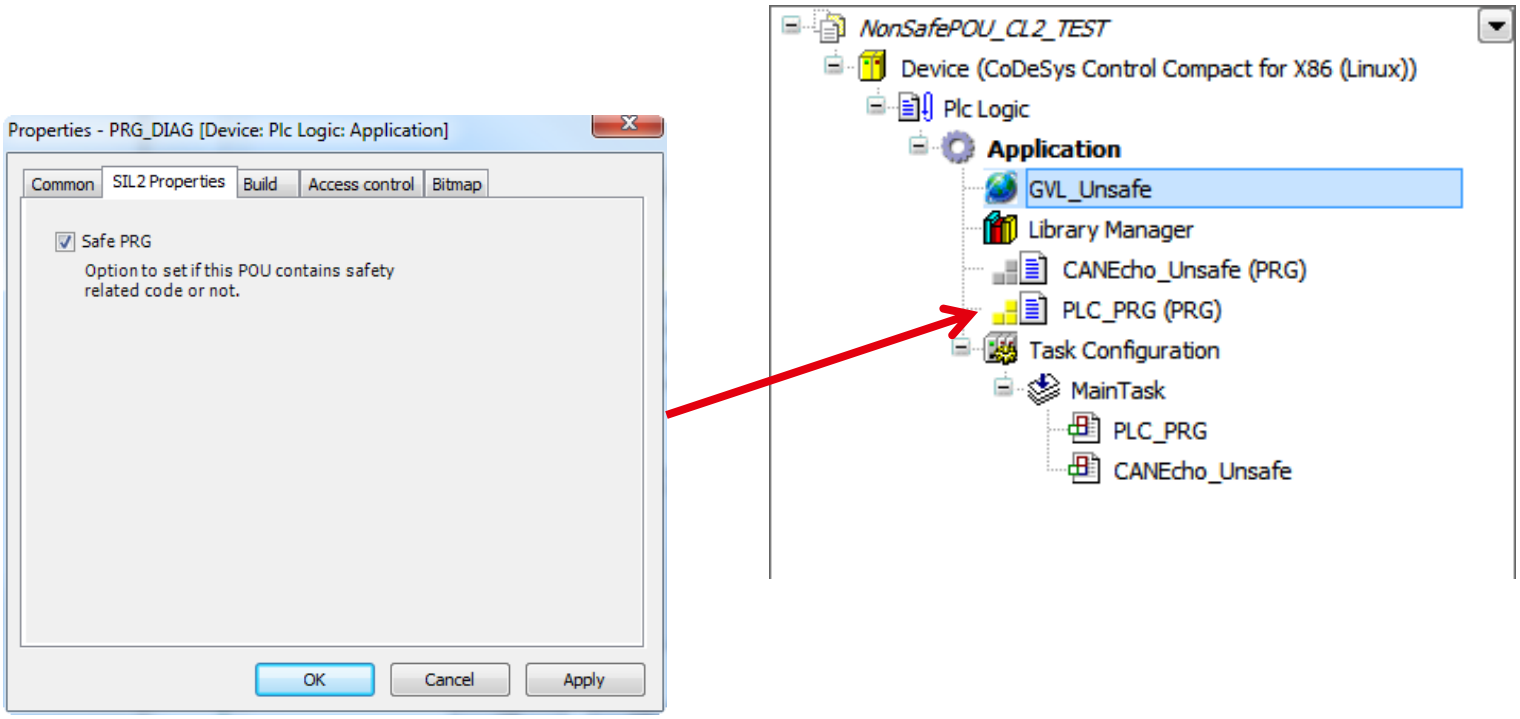
CODESYS Safety SIL2

- Combination of safe and unsafe applications
  - Data exchange, using global variable lists – synchronized by CODESYS



CODESYS Safety SIL2

- Combination of safe and unsafe applications
  - Other possibility: non-safe PRGs



  - Safety application may also contain non-safe PRGs (combination possible)

Agenda

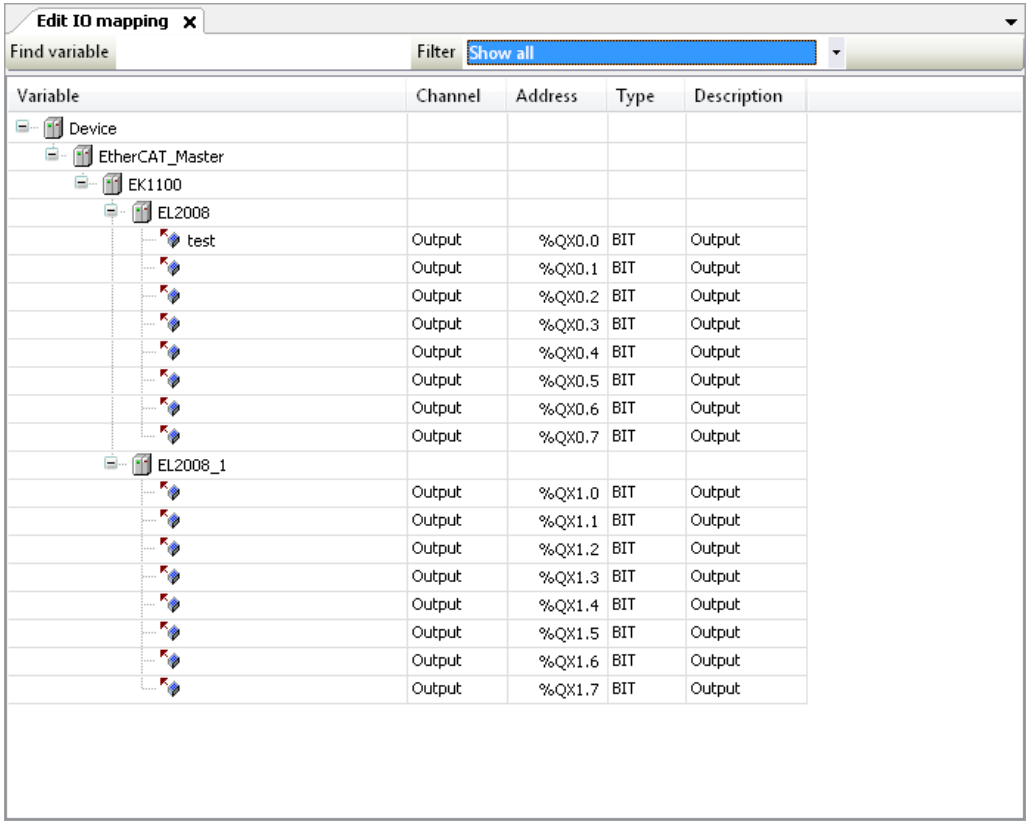
Fieldbus

Overview

- General improvements
- EtherCAT
- CANopen
- Ethernet/IP scanner stack

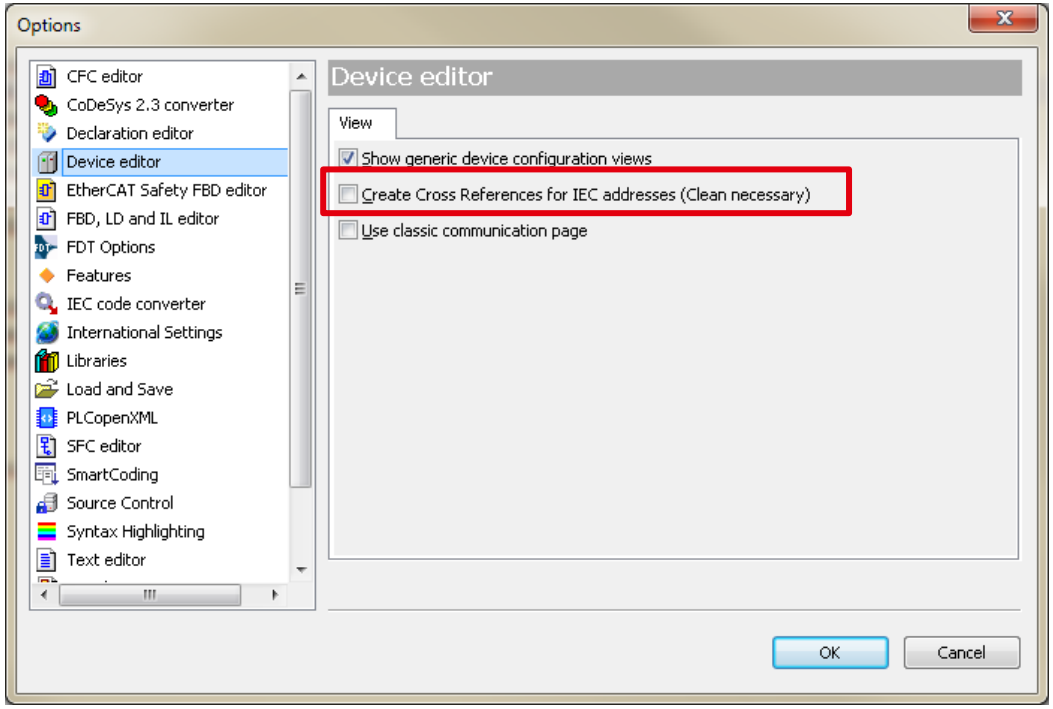
General improvements

- I/O mapping editor: Easy input of all mappings of a configuration with extra editor



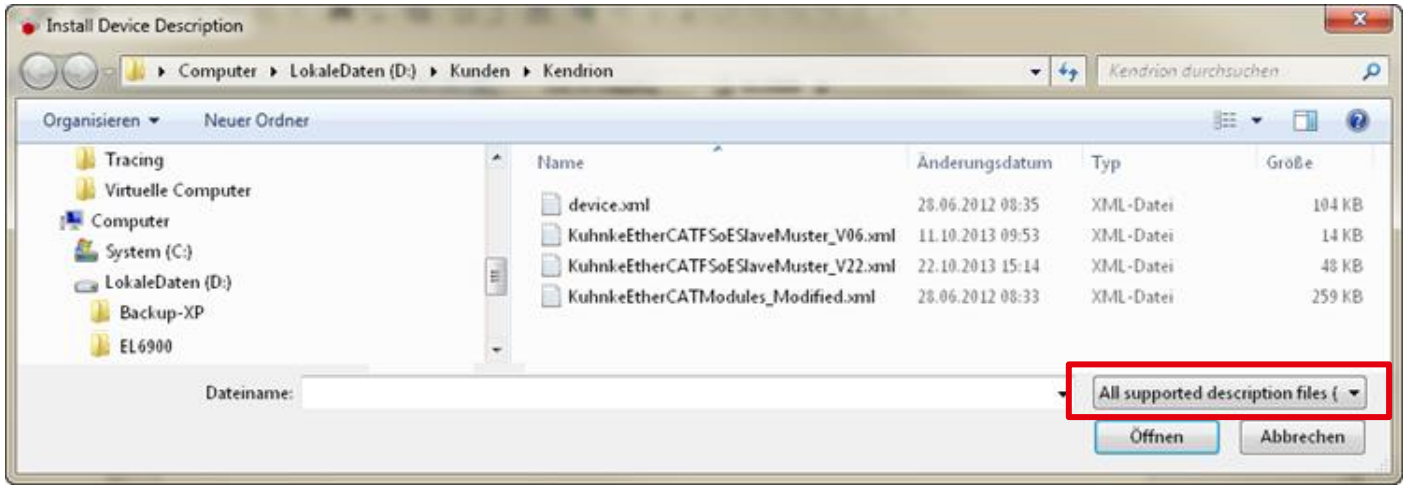
General improvements

- Search/replace and cross reference finds unmapped I/Os
  - Activation in the options for cross references necessary: requires additional time for large configurations



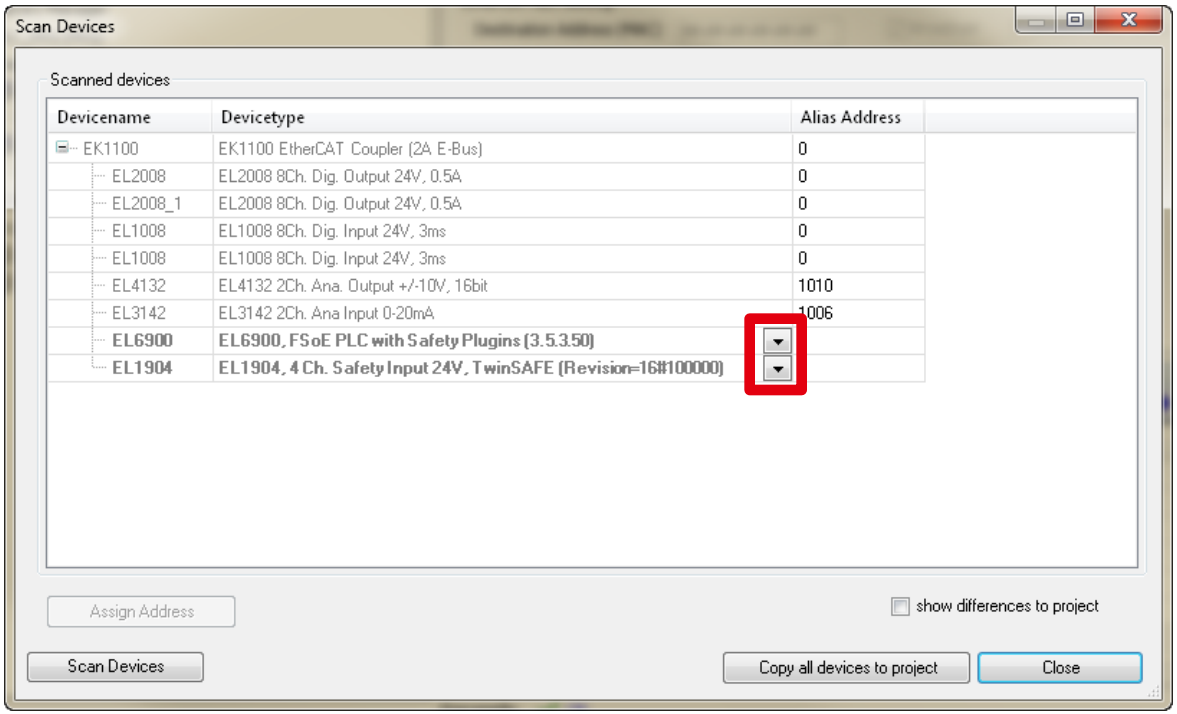
General improvements

- Multi select and multi copy for channels in the IO tab
  - Functionality identical with I/O mapping editor
  - Line end with <RETURN>: direct change to next line
- Device repository: determine file types automatically
  - Automatic detection/installation of all xml, CANopen eds and Profibus files



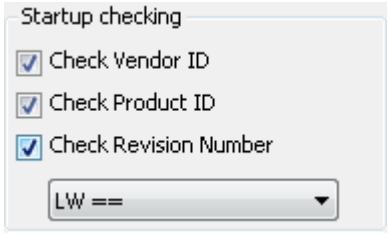
General improvements

- Device Scan ambiguous scan results: display icon on the side
  - Several options marked more clearly

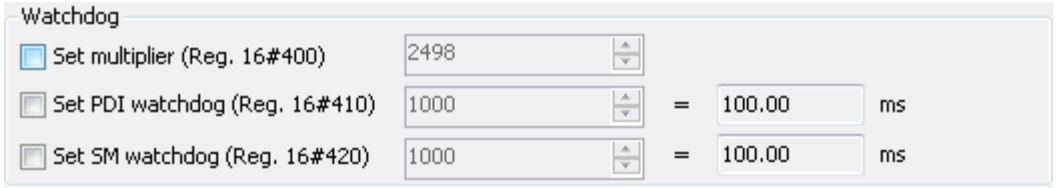


CODESYS EtherCAT

- EtherCAT add check revision and check with several options

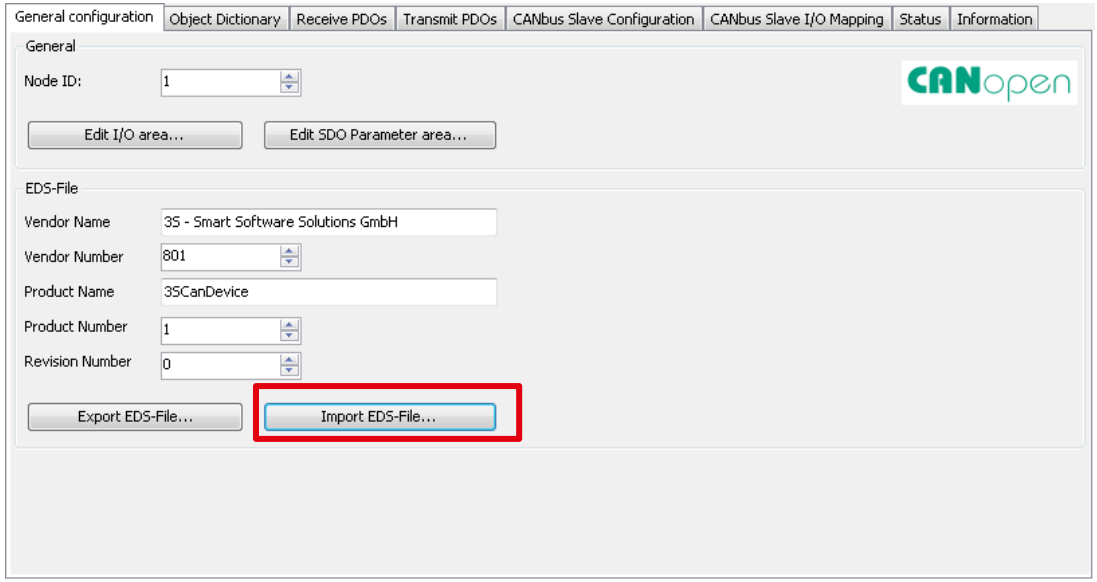


- EtherCAT Expert mode for watchdog settings



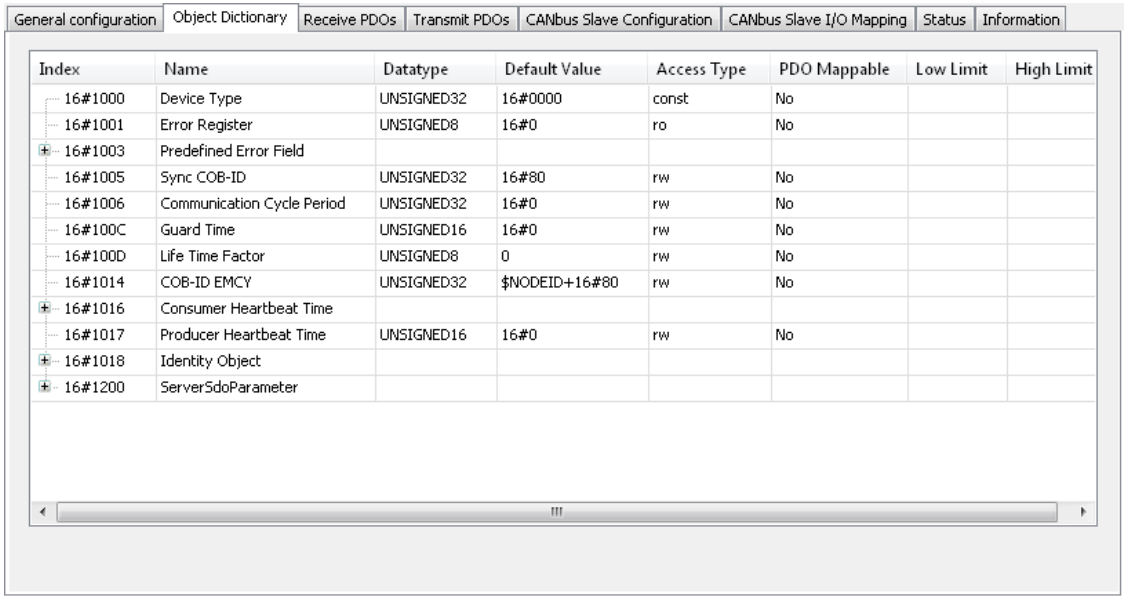
CODESYS CANopen

- CANopen Safety Slave Stack released
- CANopen Slave: Possibility to import configuration from EDS file
  - Although functionality was developed for CANopen Safety, general usage possible



CODESYS CANopen

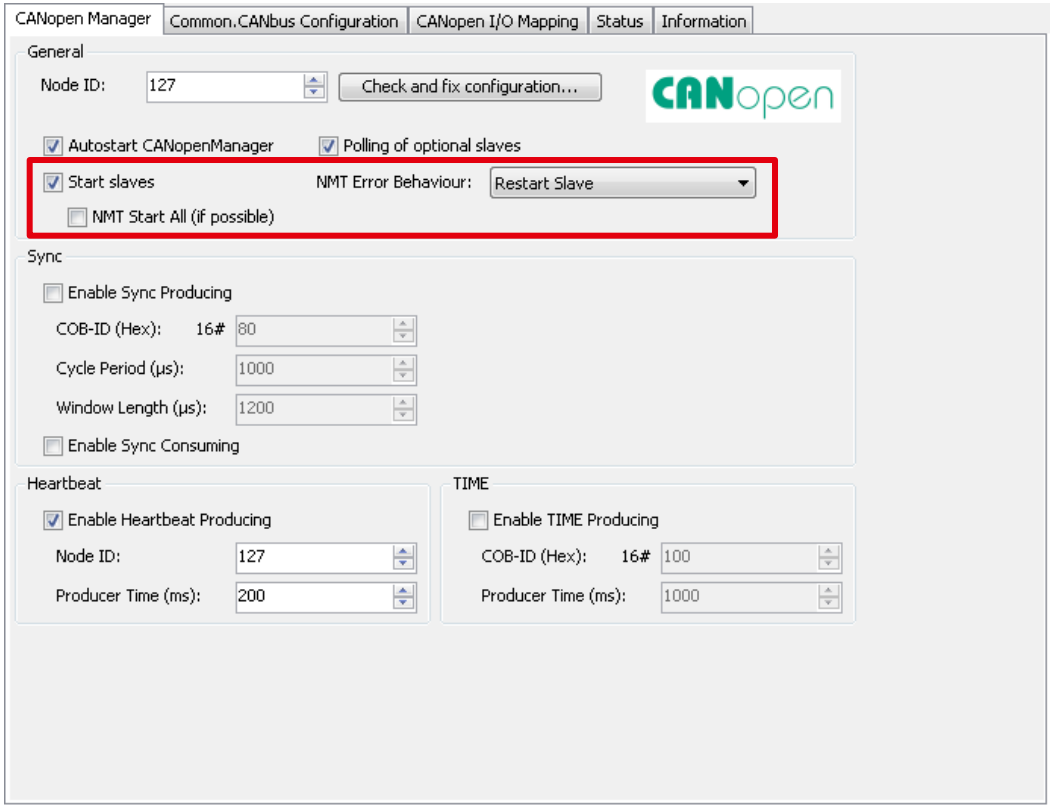
- CANopen Slave: Expert version with completely configurable object dictionary
  - Free definition of objects with index >= 0x2000, definition of other objects through EDS import
  - API extensions in the stack: read/write of the object directory per IEC code



Index	Name	Datatype	Default Value	Access Type	PDO Mappable	Low Limit	High Limit
16#1000	Device Type	UNSIGNED32	16#0000	const	No		
16#1001	Error Register	UNSIGNED8	16#0	ro	No		
16#1003	Predefined Error Field						
16#1005	Sync COB-ID	UNSIGNED32	16#80	rw	No		
16#1006	Communication Cycle Period	UNSIGNED32	16#0	rw	No		
16#100C	Guard Time	UNSIGNED16	16#0	rw	No		
16#100D	Life Time Factor	UNSIGNED8	0	rw	No		
16#1014	COB-ID EMCY	UNSIGNED32	$NODEID+16#80	rw	No		
16#1016	Consumer Heartbeat Time						
16#1017	Producer Heartbeat Time	UNSIGNED16	16#0	rw	No		
16#1018	Identity Object						
16#1200	ServerSdoParameter						

CODESYS CANopen

- CANopen: Specifying NMT error event behaviour now possible



CODESYS Ethernet/IP

- Software protocol stack for Ethernet/IP Scanner (Master) now available as portable CODESYS library (realized in IEC 61131-3)
  ➔ no more additional hardware needed
- Hilscher CIFx card will continue to be supported
- Ethernet/IP user library for acyclic services
  - Function blocks and visualization templates for access to Ethernet/IP adapter (=Slave) from the IEC application

Inspiring Automation Solutions

Thank you for your attention.