Overview

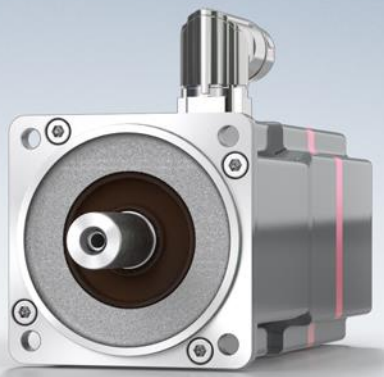
BECKHOFF

PC-based Control

BECKHOFF

IPC

I/O



Motion

Automation

Contents

BECKHOFF

PC-based Control – Milestones

BECKHOFF

1988

S1000
PLC/NC on
PC (DOS)

PC-based Control – Milestones

BECKHOFF

1988

S1000
PLC/NC on
PC (DOS)

1993

S2000
PLC/NC/CNC
on PC

PC-based Control – Milestones

BECKHOFF

1988

S1000
PLC/NC on
PC (DOS)

1993

S2000
PLC/NC/CNC
on PC

1996

TwinCAT
IEC 61131
PLC real-time under
Windows NT

PC-based Control – Milestones

BECKHOFF

1988

S1000
PLC/NC on
PC (DOS)

1993

S2000
PLC/NC/CNC
on PC

1996

TwinCAT 2
IEC 61131
PLC real-time under
Windows NT

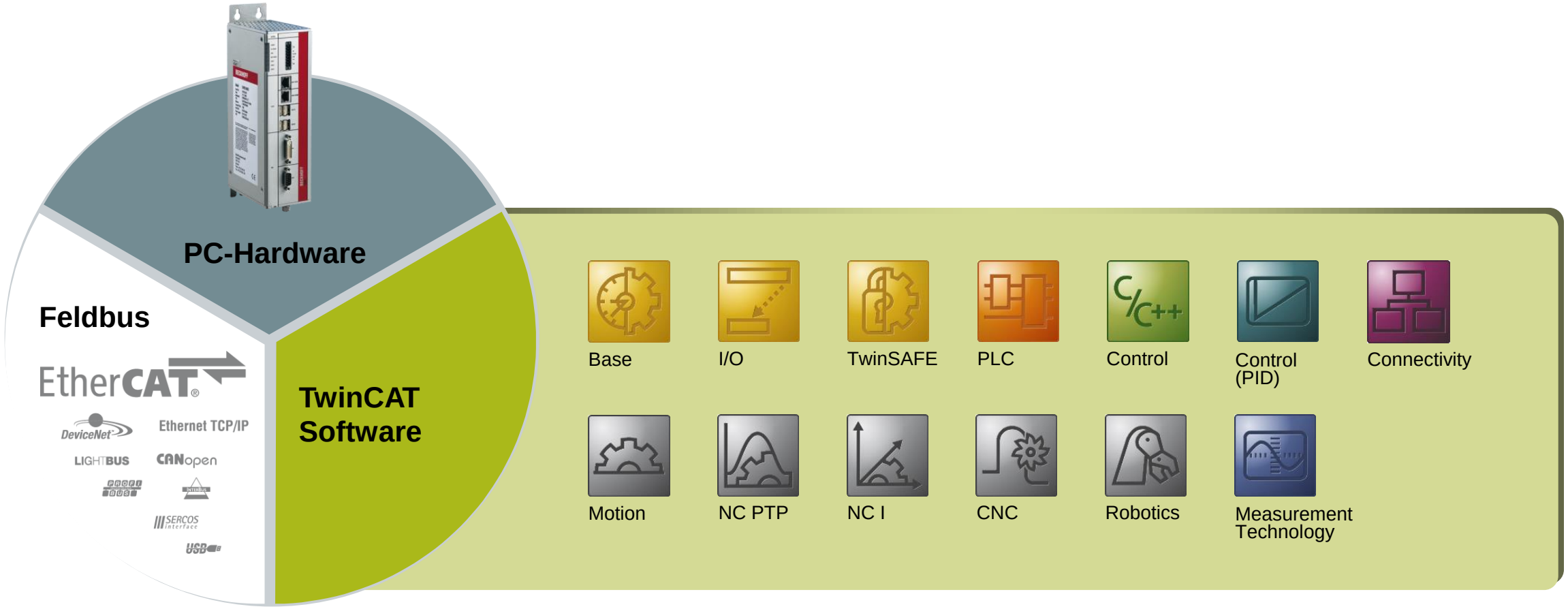
2010

TwinCAT 3

Motivation

BECKHOFF

Beckhoff PC-based Control:
Setting New Standards in Automation



Motivation

BECKHOFF

- integration of PLC, Motion and HMI into **one** software on **one** CPU:
  - minimised hardware
  - faster cycle times due to no hardware interfaces
  - reduced interface complexity
  - better and easier diagnosis

- PC Control offers an “open” control system
  - abstraction as a principle
  - functions in software & independent from hardware

Motivation

BECKHOFF

- scalable performance and cost by selection of standard CPUs
- powerful operating systems add IT functions to automation solutions
- automation and IT world share the same benefits:
  - performance increase
  - cost decrease

Contents

BECKHOFF

eXtended Automation (XA)

BECKHOFF

- Architecture (XAA)
- Engineering (XAE)
  - System Manager
  - PLC
  - Motion Control
  - C/C++ programming
  - MATLAB® integration
  - C#/.NET programming
- Runtime (XAR)

Architecture (XAA)

BECKHOFF

eXtended Automation Technology (XAT)

- More than standard automation

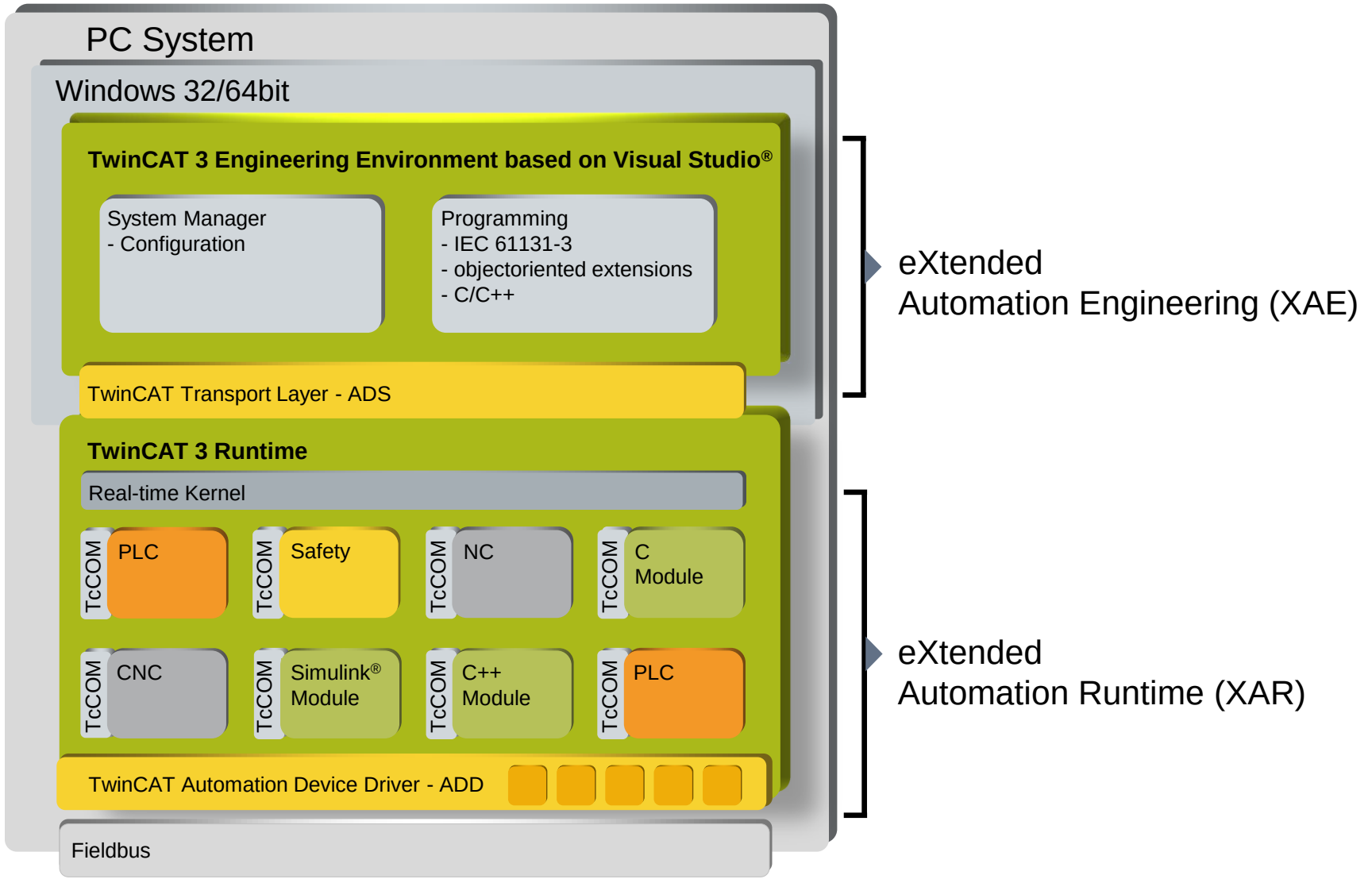
eXtended Automation Engineering (XAE)

- Visual Studio® for programming in IEC 61131-3, C/C++
- Visual Studio® for configuring the complete system (integrated TwinCAT System Manager)

eXtended Automation Runtime (XAR)

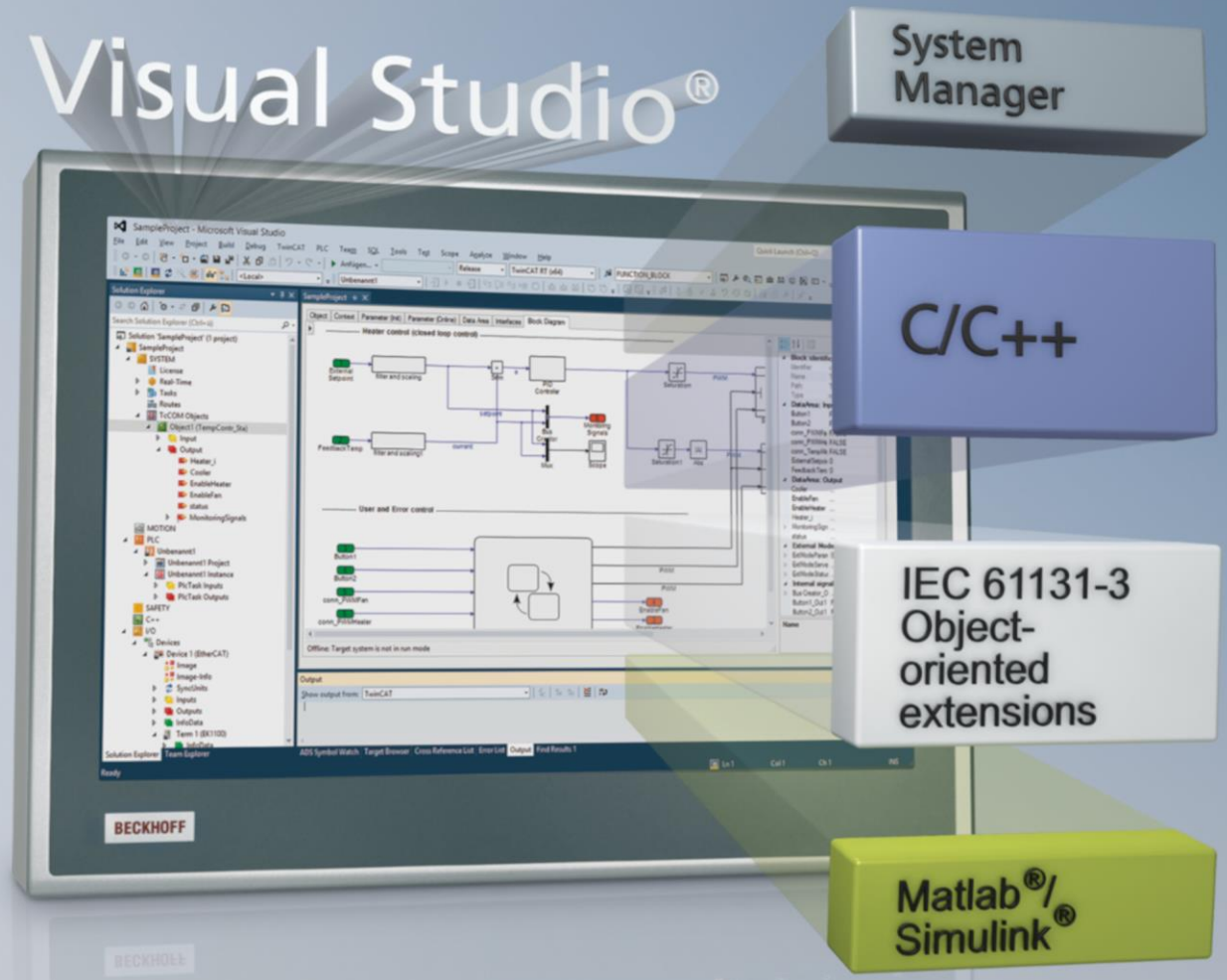
- Executes modules in real-time
- Supports multi-core CPUs
- Supports 64-bit operating systems

Architecture (XAA)



Engineering (XAE)

BECKHOFF



System Manager

C/C++

IEC 61131-3 Object-oriented extensions

Matlab®/ Simulink®

TwinCAT® 3

Features

BECKHOFF

- TwinCAT 3 – extended, modular engineering tool
- **one** programming environment, **one** project file, **one** debug environment
- **integrated** TwinCAT System Manager
- programming according to IEC 61131-3 3rd edition
  (including NEW object orientation extensions)

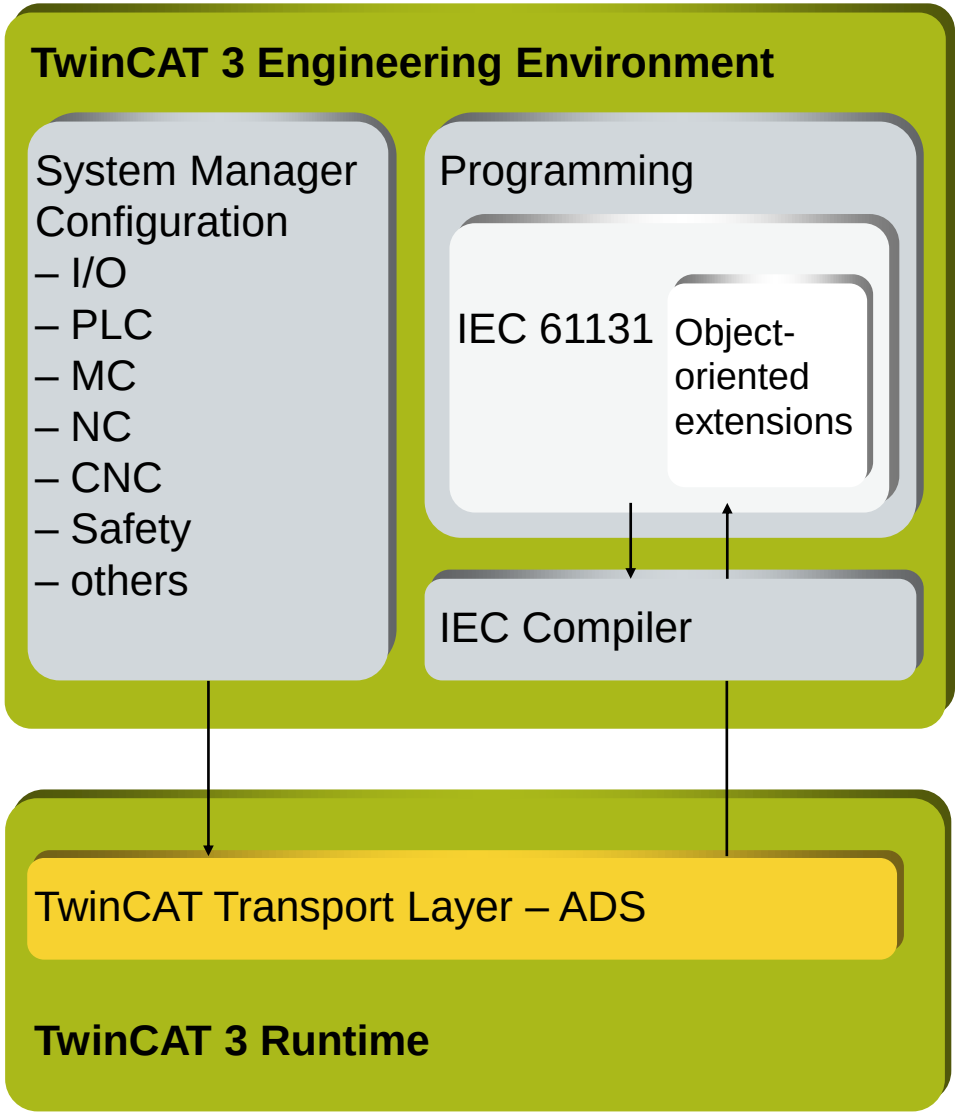
- Optional usage of C and C++ for real-time programming
- link to MATLAB®/Simulink®
- runs all TwinCAT 2 PLC projects without change or migration of TwinCAT 2 projects (conversion)
- based on Microsoft Visual Studio®

Scalable functionality

BECKHOFF

TwinCAT 3 Standard

- Using XAE Shell (based on Microsoft Visual Studio®)
- for PLC programmers and users of existing modules
- configuring, setting up and diagnostics
- debugging PLC



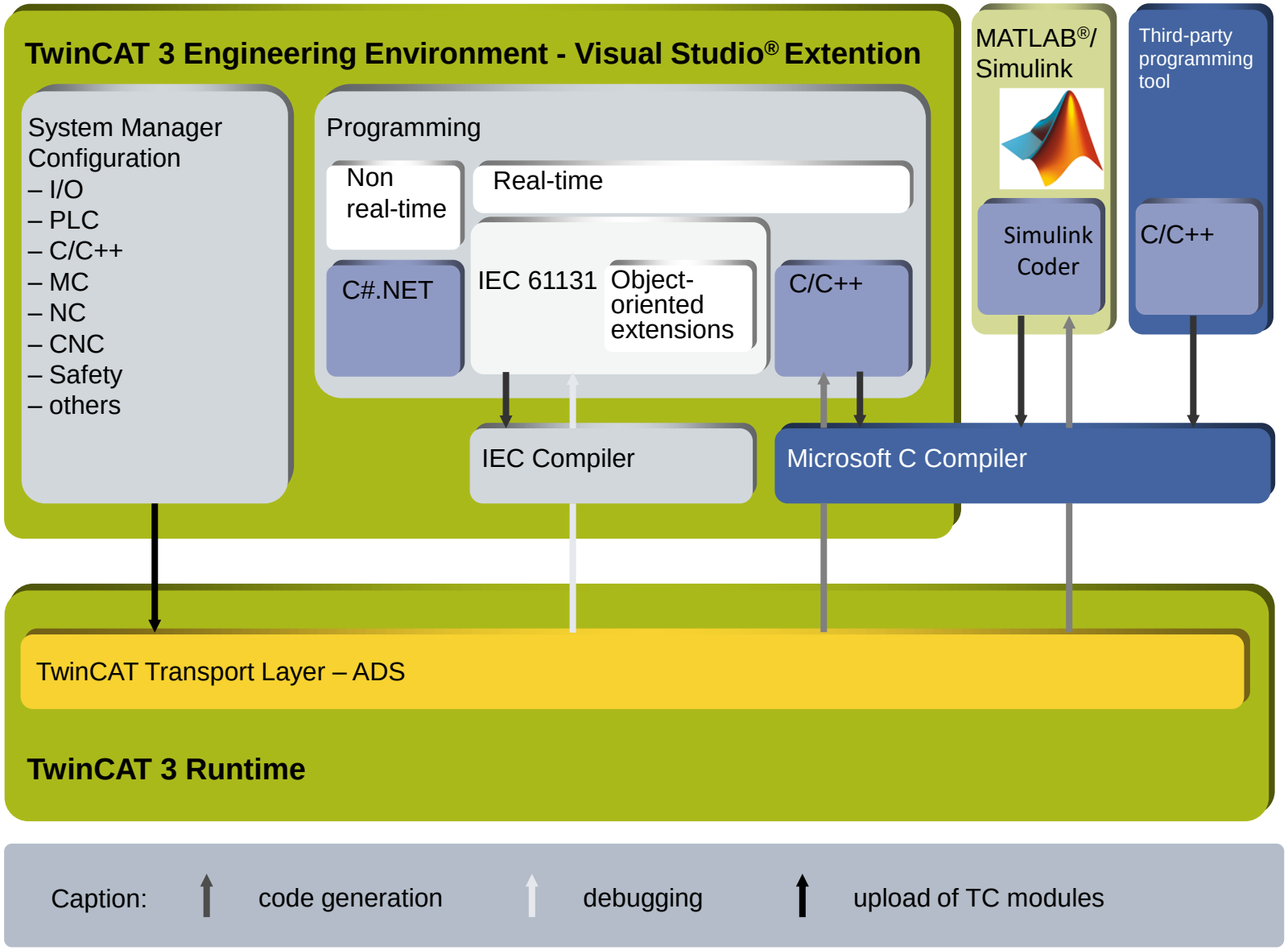
Scalable functionality

BECKHOFF

TwinCAT 3 Integrated

- Integrated into Microsoft Visual Studio®
- for PLC and C/C++ programmers
- configuring, setting up and diagnosis
- module generation (C/C++, MATLAB®/Simulink®)

- debugging
  - PLC
  - C/C++
  - MATLAB®/Simulink®



Workbench integration

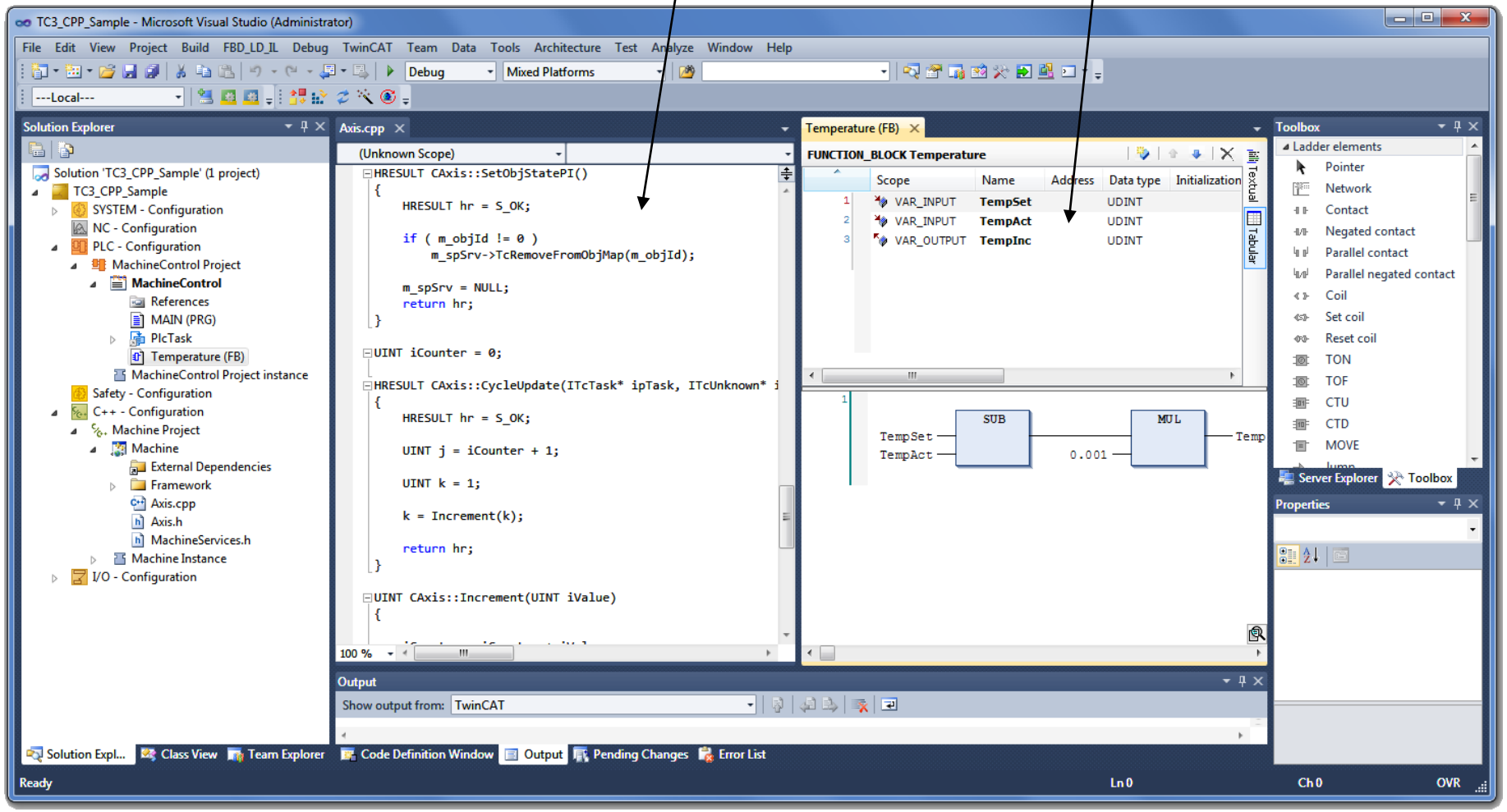
TwinCAT 3 framework = Microsoft Visual Studio®

- usage of the most common programming environment
- maintained by a company
- extendable via plug-ins
- Integration to source control software
- usage of C and C++ for programming automation devices
- usage of .NET languages for non-real-time applications (e.g. HMI)
- standardized Help system

Free choice of the programming language

BECKHOFF

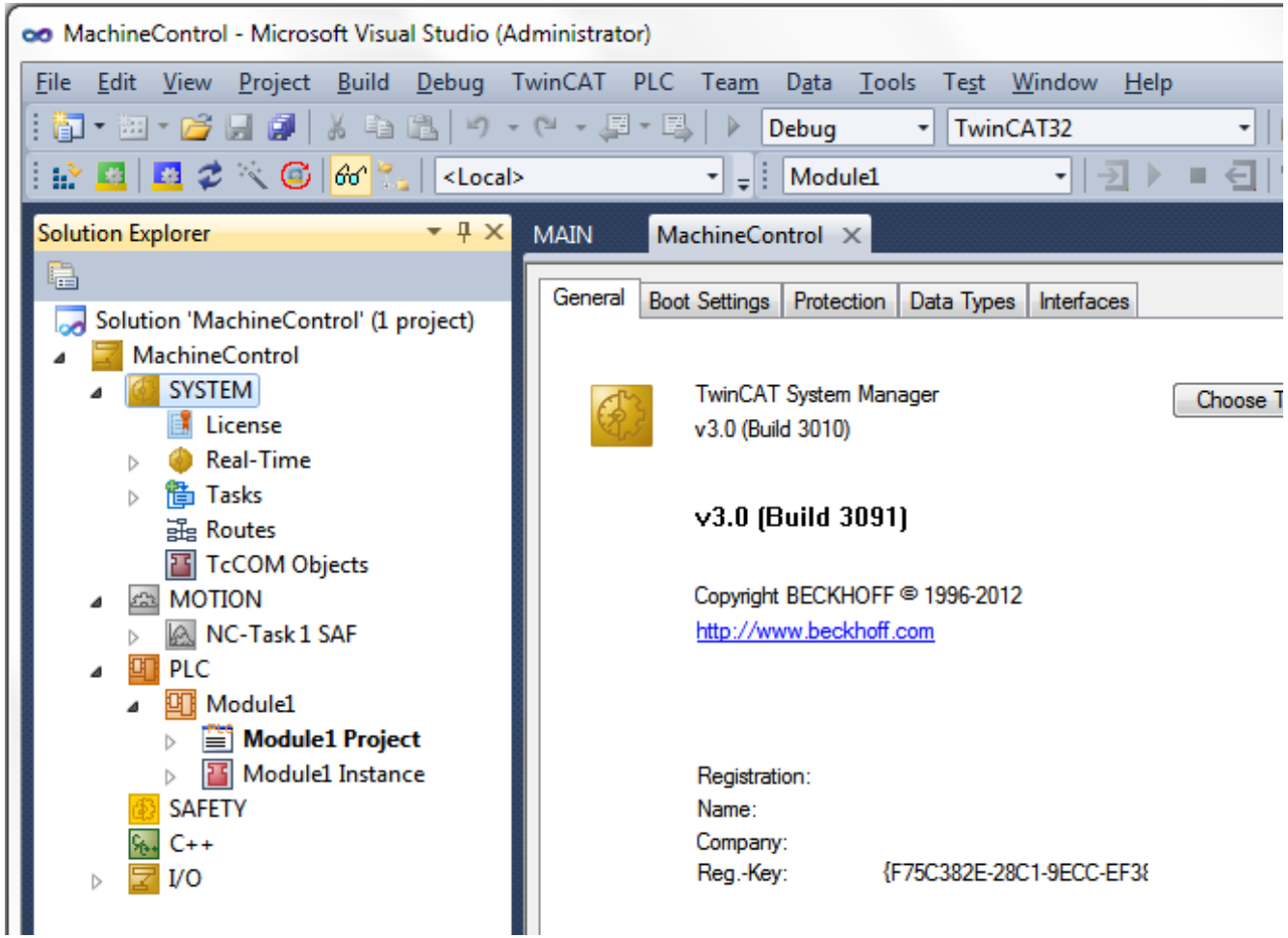
Simultaneous programming of C++ and IEC 61131-3



TwinCAT I/O – Integrated System Manager

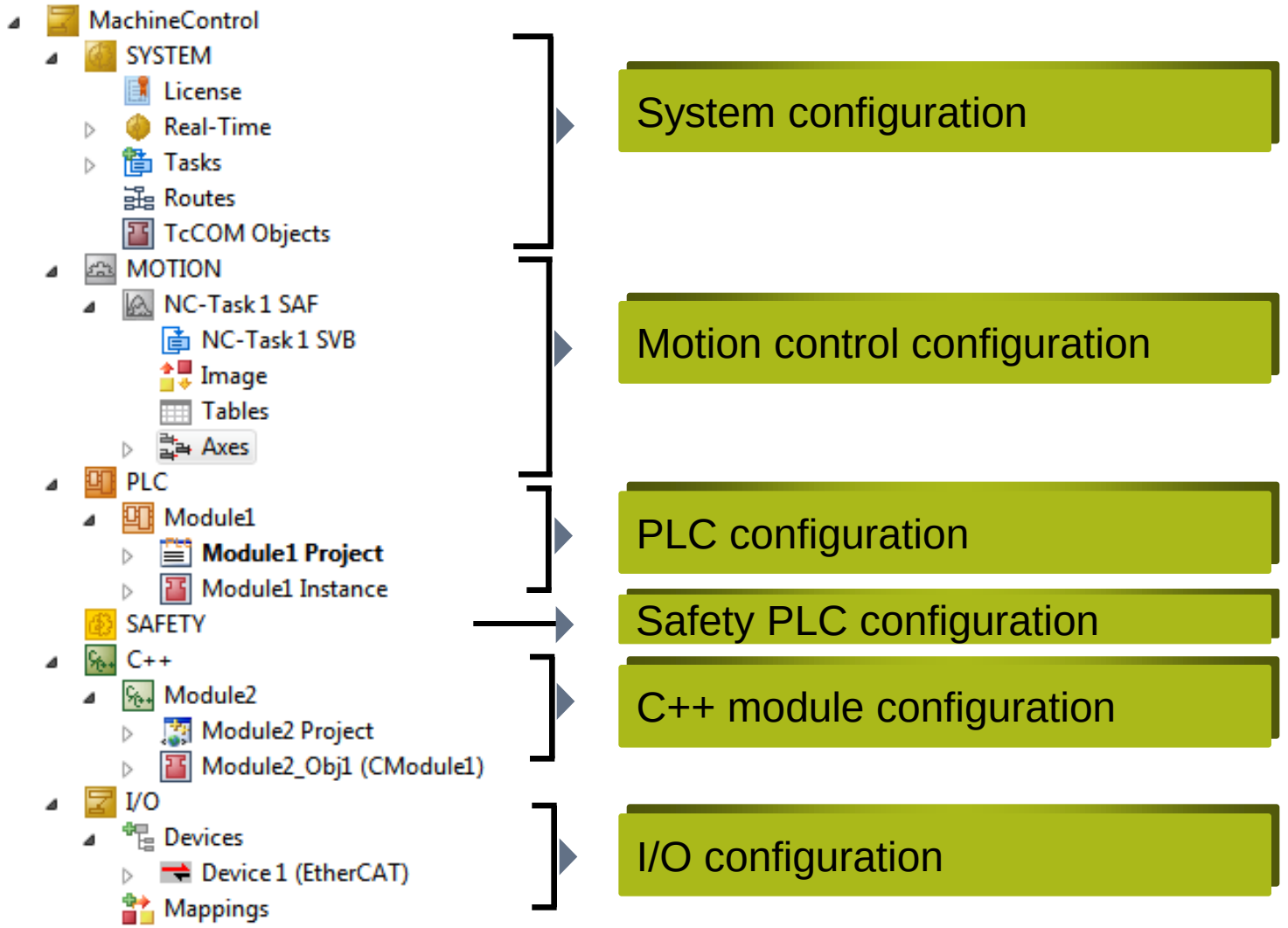
BECKHOFF

- One tool for:
  - programming, configuration and diagnosis
- uniform task management
- parameterization of TwinCAT modules
- creation and administration of mappings between all process images
- simulation of I/Os and Axes



TwinCAT I/O – Integrated System Manager

BECKHOFF

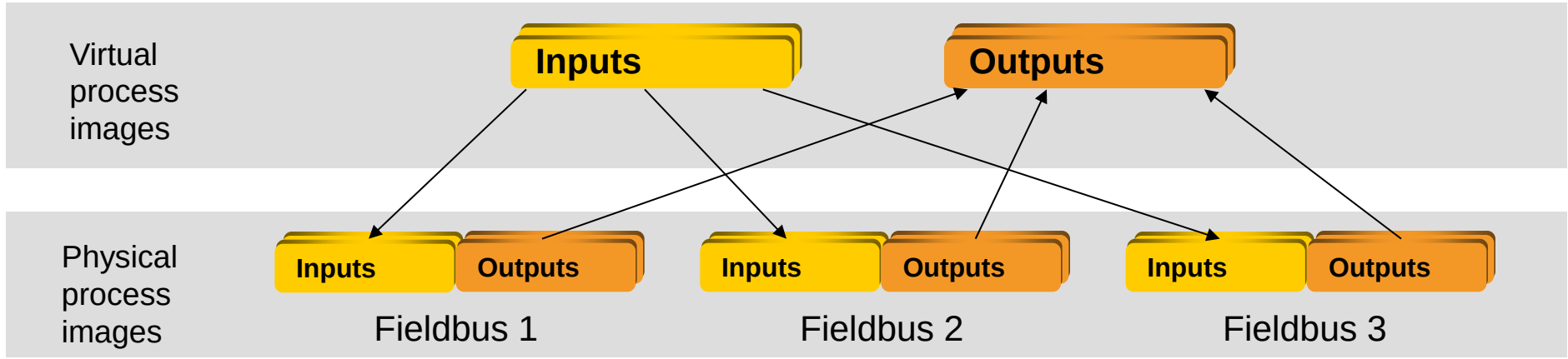


Mapping of process images

- open for all common fieldbuses
- support of all PC hardware interfaces
- easy commissioning and diagnosis
- assignment of logical and physical process images

→ Changes of bus system do not require changes to PLC code.

TwinCAT process images



PLC projects:

- Unlimited Projects per PC - only limited by memory (instead of 4 in TwinCAT 2)
- PLC Task 65,000 possible per PLC (instead of 4 in TwinCAT 2)

Programming:

- languages of the IEC 61131-3 (IL, ST, FBD, LD, SFC) + CFC
- usage of the object-oriented extensions of the 3rd edition of the IEC 61131
- call and data exchange of modules written in C/C++ and MATLAB®/Simulink®
- multiple import and export interfaces
- no direct addressing necessary

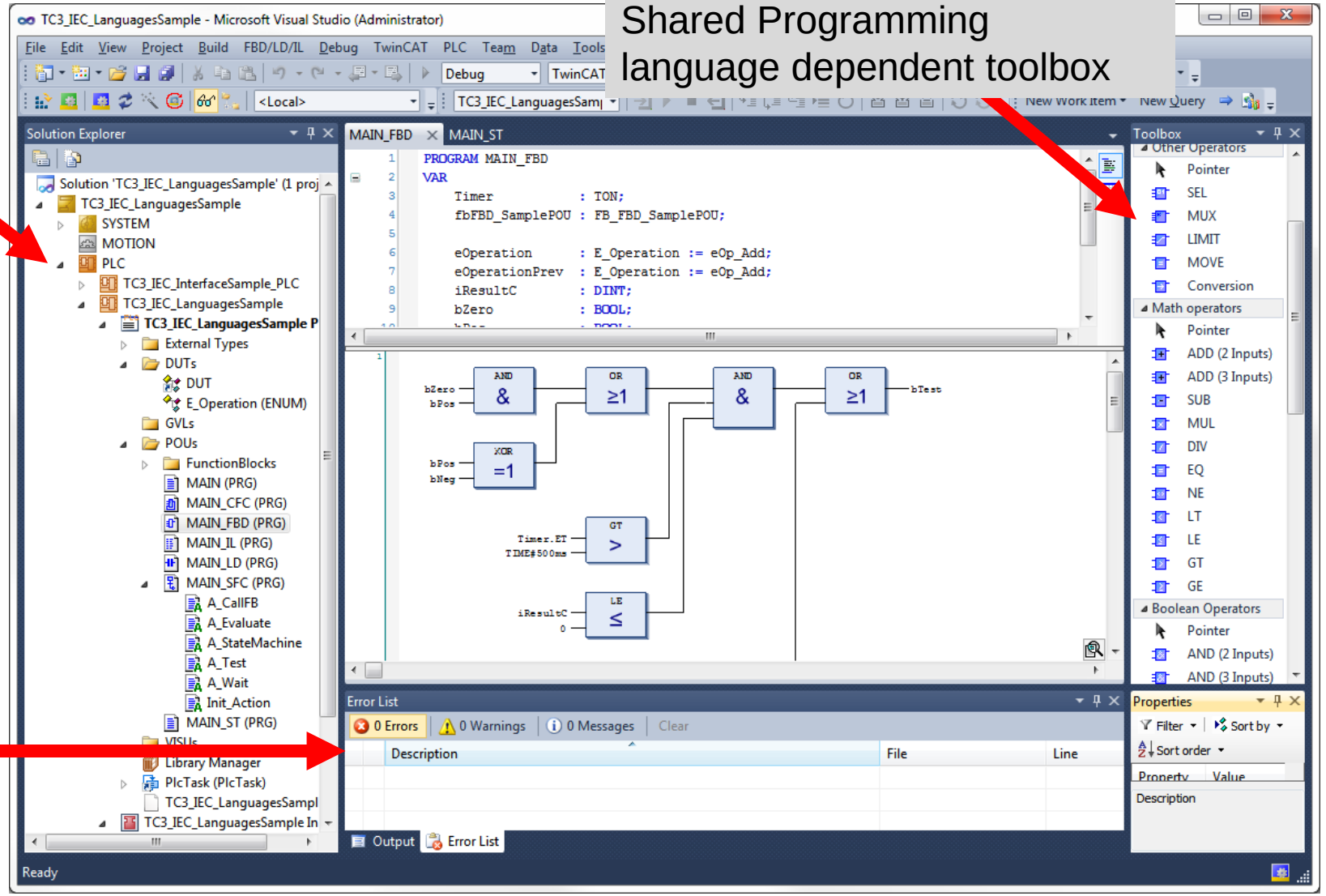
Commissioning/maintenance:

- source code upload and download
- online change
- full debugging functionality (breakpoints, monitoring, flow control,…)

Shared tree structure for software and hardware config

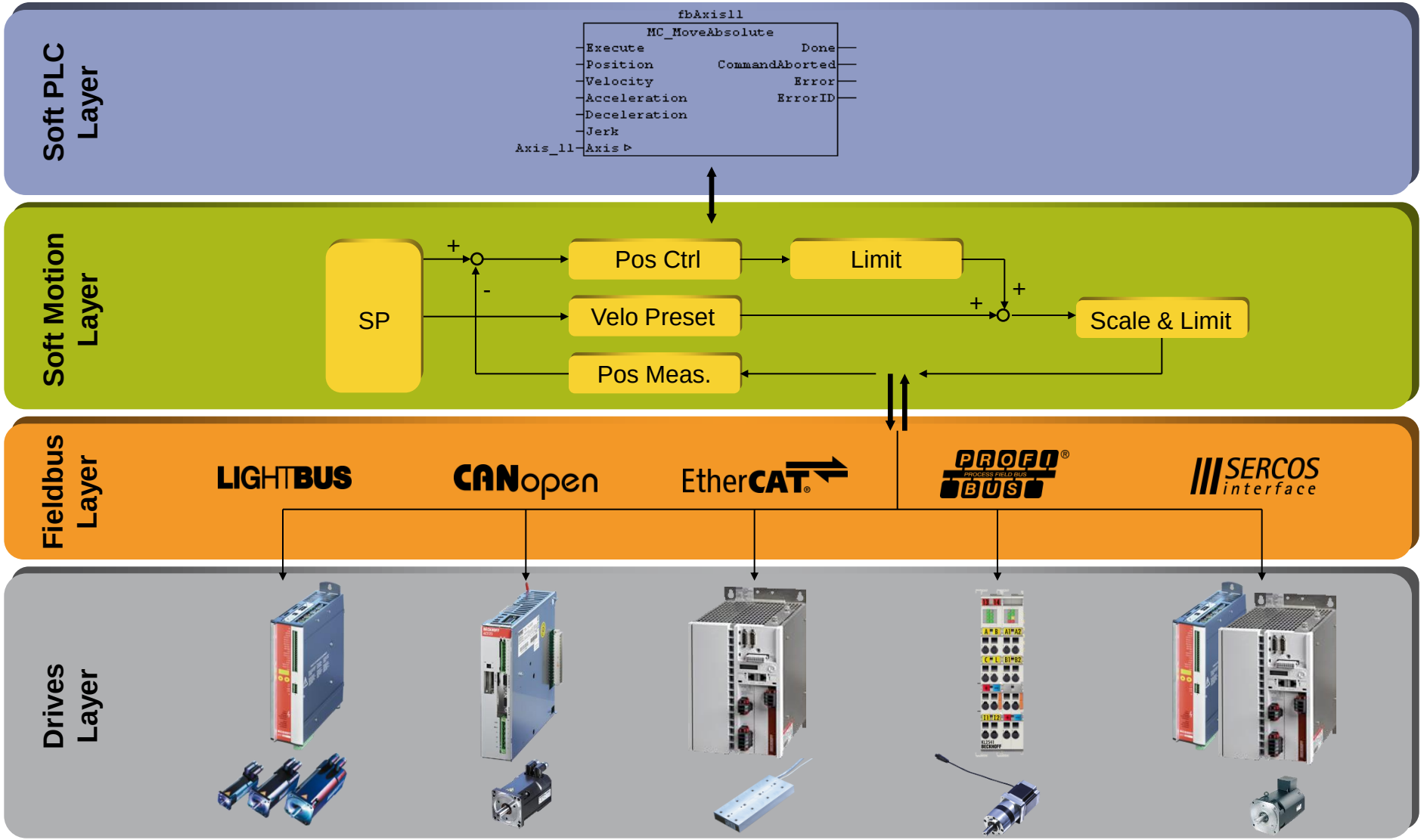
Shared Programming language dependent toolbox

Universal output window for feedback from all Projects



TwinCAT 3 Motion Control – Overview

BECKHOFF



Functionality

NC PTP

Point-to-Point Movement
- Gearing
- Camming
- Superposition
- Flying saw

PLCopen motion control

NC I

Interpolated motion
- 3 axes interpolated with 5 additional axes
- Native G-Code execution
- Advanced Features
- straightforward utilization through function blocks from the PLC

CNC

Complete CNC functionality
- interpolated movement for up to 32 axes per channel
- various transformations

Robotics

Interpolated motion for robotic control
- support for a wide range of kinematic systems
- optional torque pre-control

TwinCAT 3 Motion Control – Cont.

BECKHOFF

- scalable solutions (stepper,…, servo drive)
- several abstraction layers
  → PLC / SCADA / HMI access always identical objects, independent of axis type and fieldbus
- conversion from mechanical to electronical system
  (electronic cam, electronic gear, electronic clutch, electronic camshaft, “flying saw”)

Benefits:

- more flexibility in used technique (stepper, servo drive,…)
- more flexibility in changes to the products
- shorter time of delivery and development time
- shorter time for commissioning because of the lack of mechanical parts
- decreasing costs

Advanced programming options

- C / C++
- MATLAB/Simulink
- C# / .NET (Non-Realtime)

C / C++

C/C++

BECKHOFF

- Allows reuse of existing C/C++ code
- cooperation of C/C++ and PLC code
- real-time applications for all platforms (Windows, TC/BSD, Linux…)
- opens new areas – not just standard PLC users
- well-known programming language
- standardised (C: ISO/IEC 9899 TC3, C++: IEC 14882)
- generation of Automation Device Drivers (ADD) enables to implement own drivers (e.g. fieldbus drivers)
- Beckhoff SDK provides the functionality (like PLC libraries) for
  - ADS
  - Motion
  - File I/O

C/C++ Programming Languages

BECKHOFF

Application areas

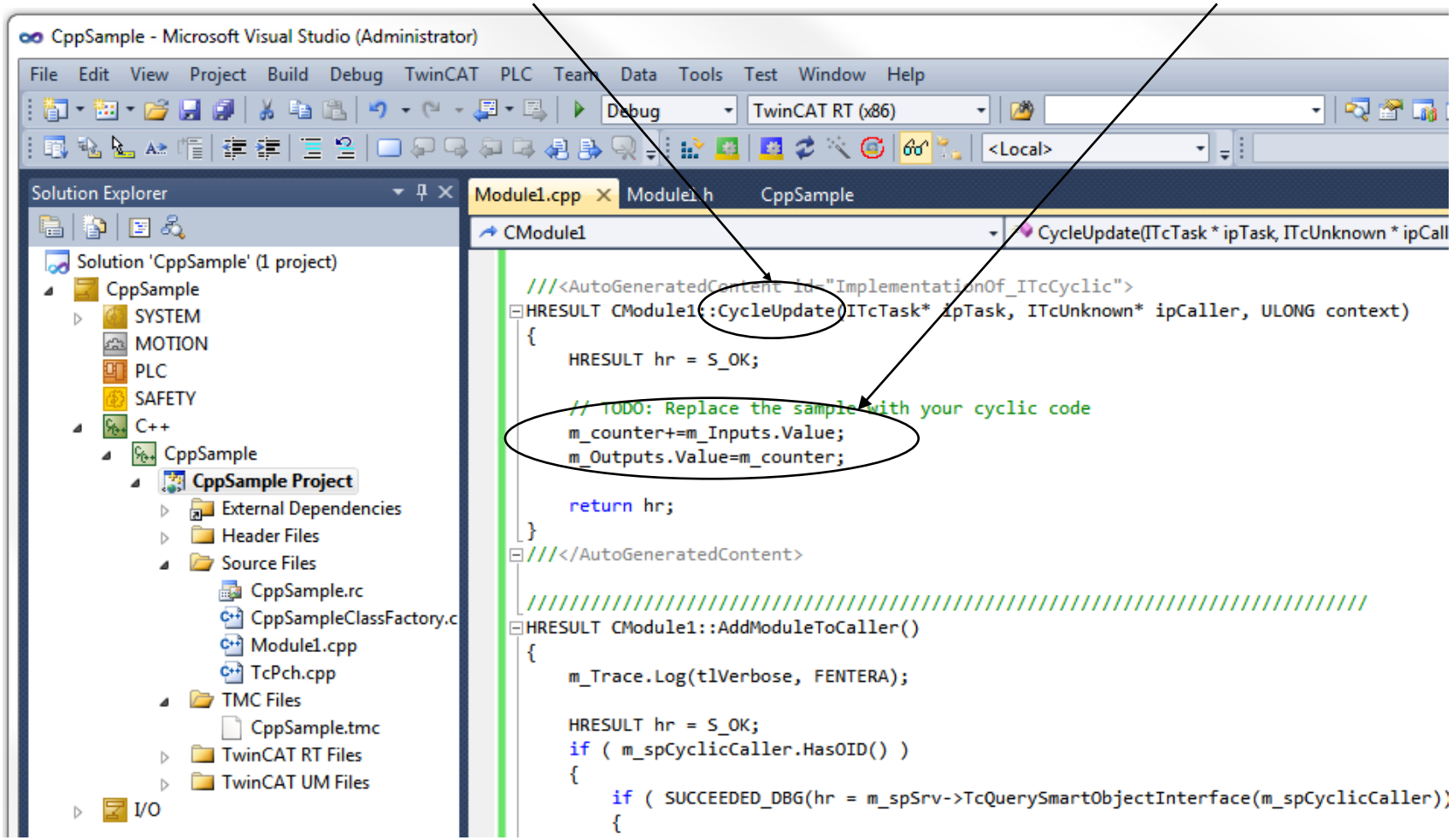
- image processing
- robotics
- measurement technology
- …

C/C++ Programming Languages

BECKHOFF

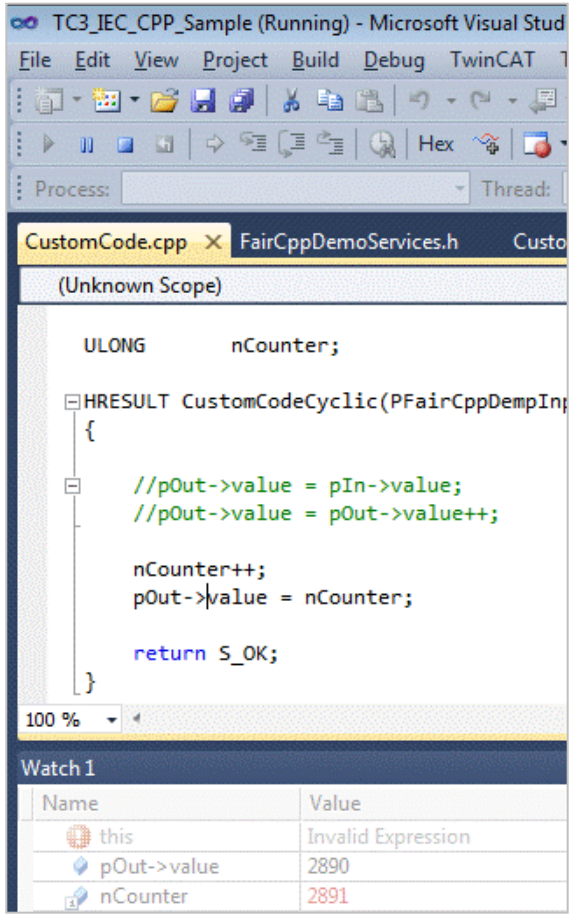
Method CycleUpdate: – is called cyclically

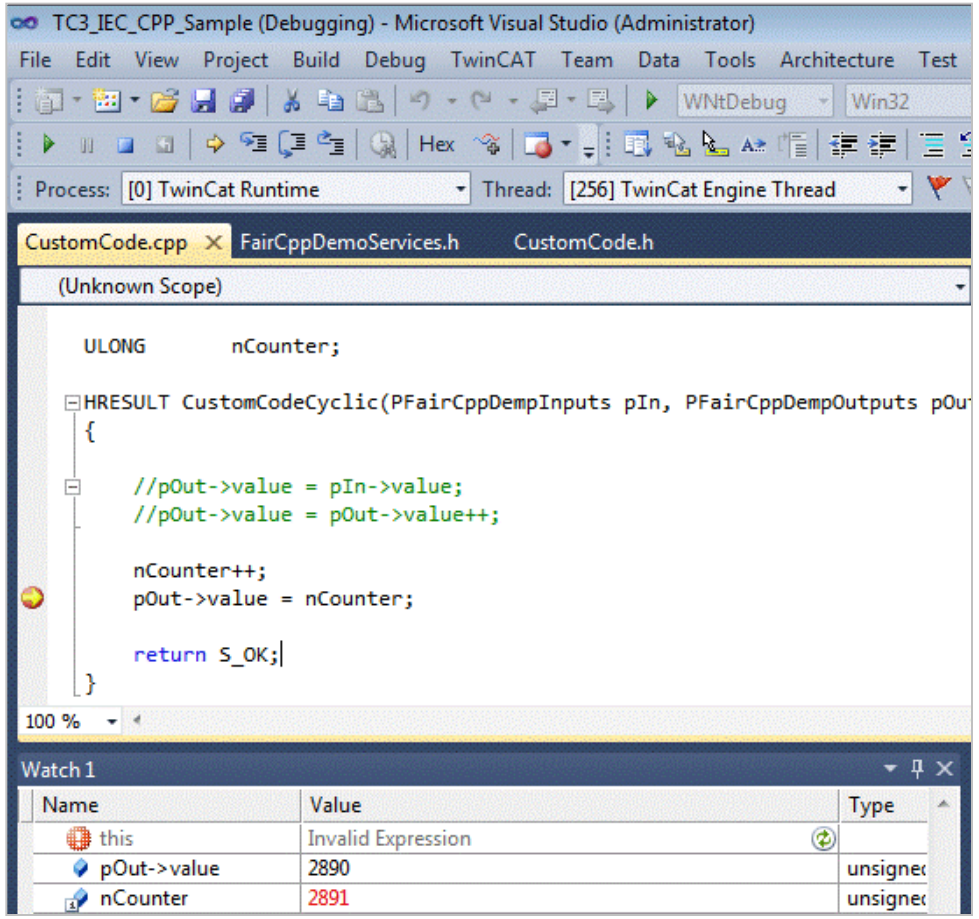
Pointer to logical input/output image



VS standard debugger:

- monitoring/modification of variables only with Breakpoint

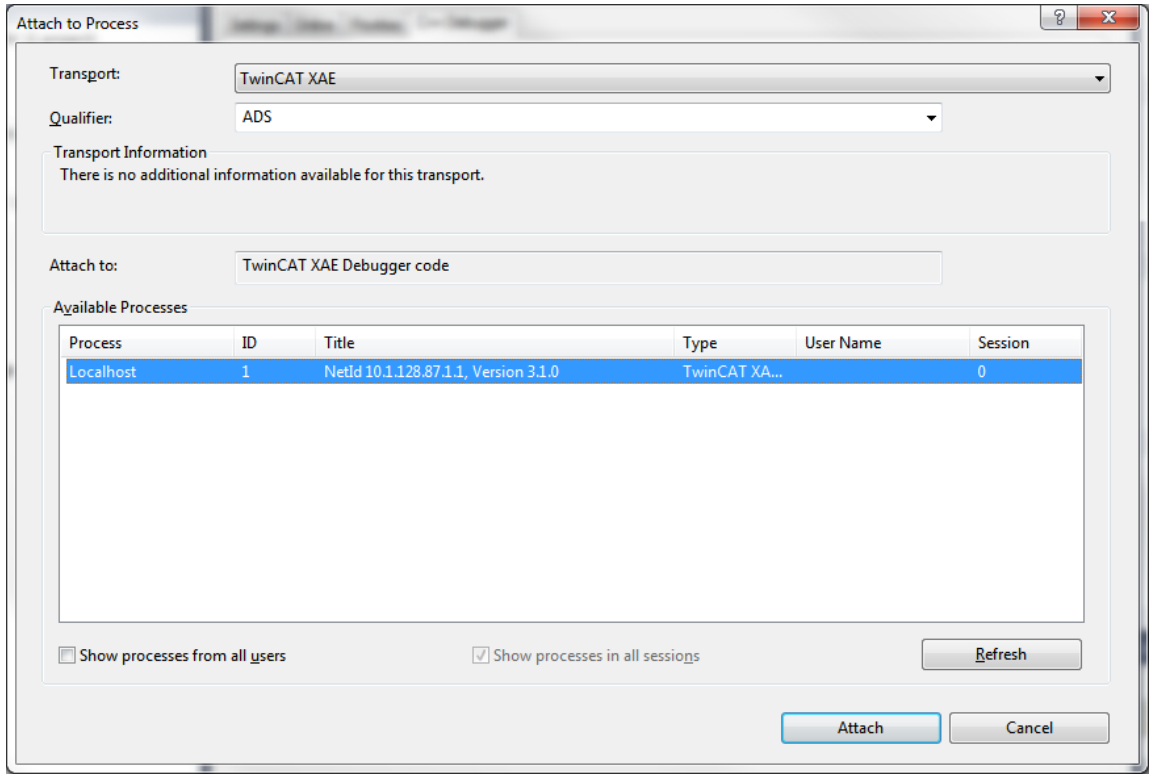




VS Beckhoff debugger:

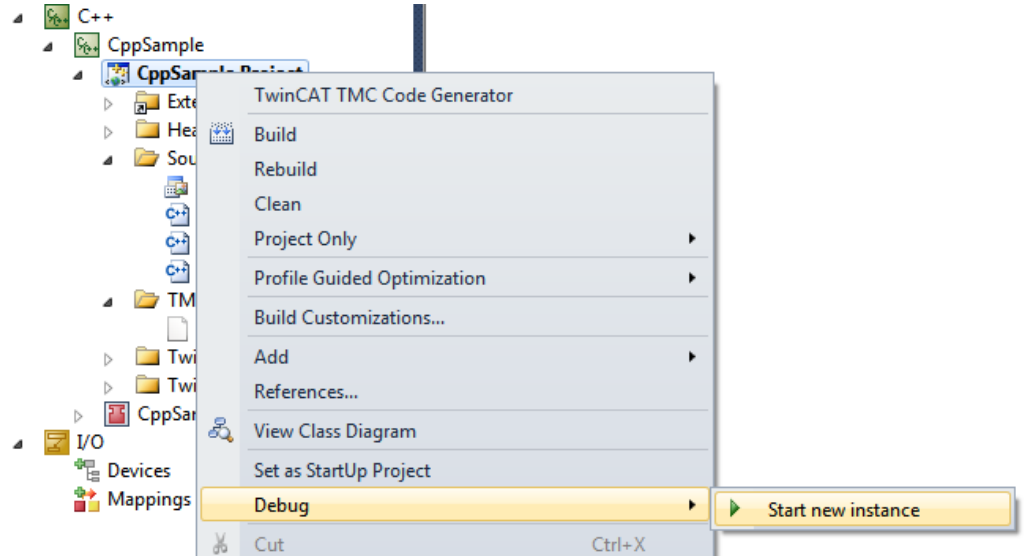
- connection to C++ target (selection of target system)

by “Attach to process”



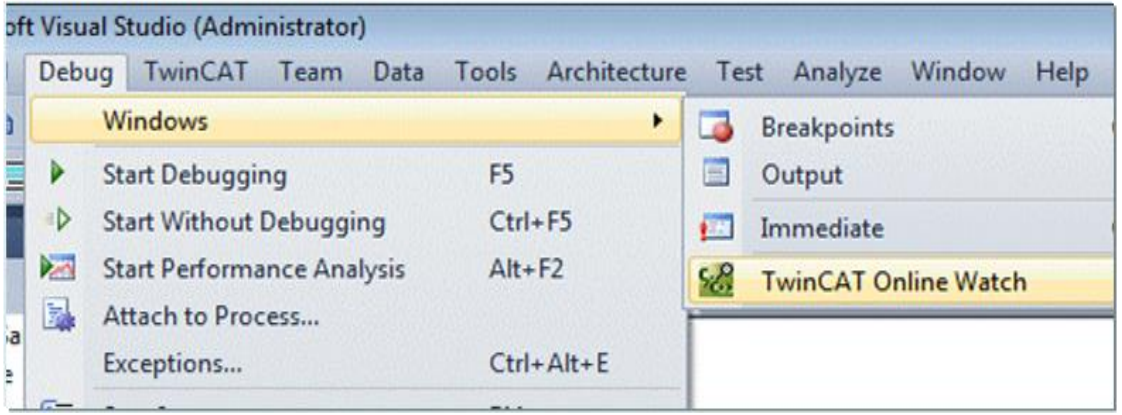
or

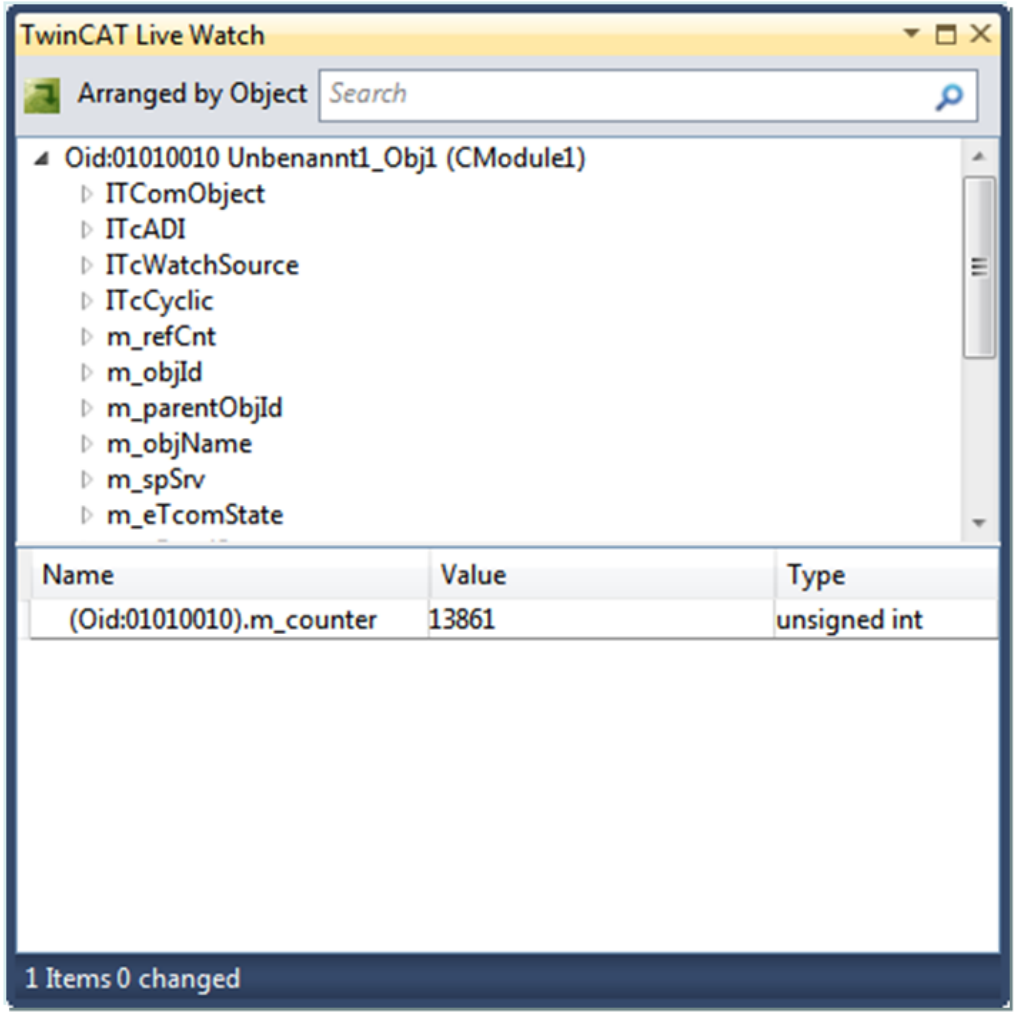
“Start new instance”



VS Beckhoff debugger:

- monitoring/modification of variables without Breakpoint (analog to PLC without Breakpoint)





MATLAB / Simulink

MATLAB ®/Simulink® integration

MATLAB®/Simulink® integration

- well-known in the scientific and measurement environment
- great variety of toolboxes (e.g. Fuzzy)
- creation, simulation and optimization of control circuits
- debug interface between Simulink® and TwinCAT

MathWorks

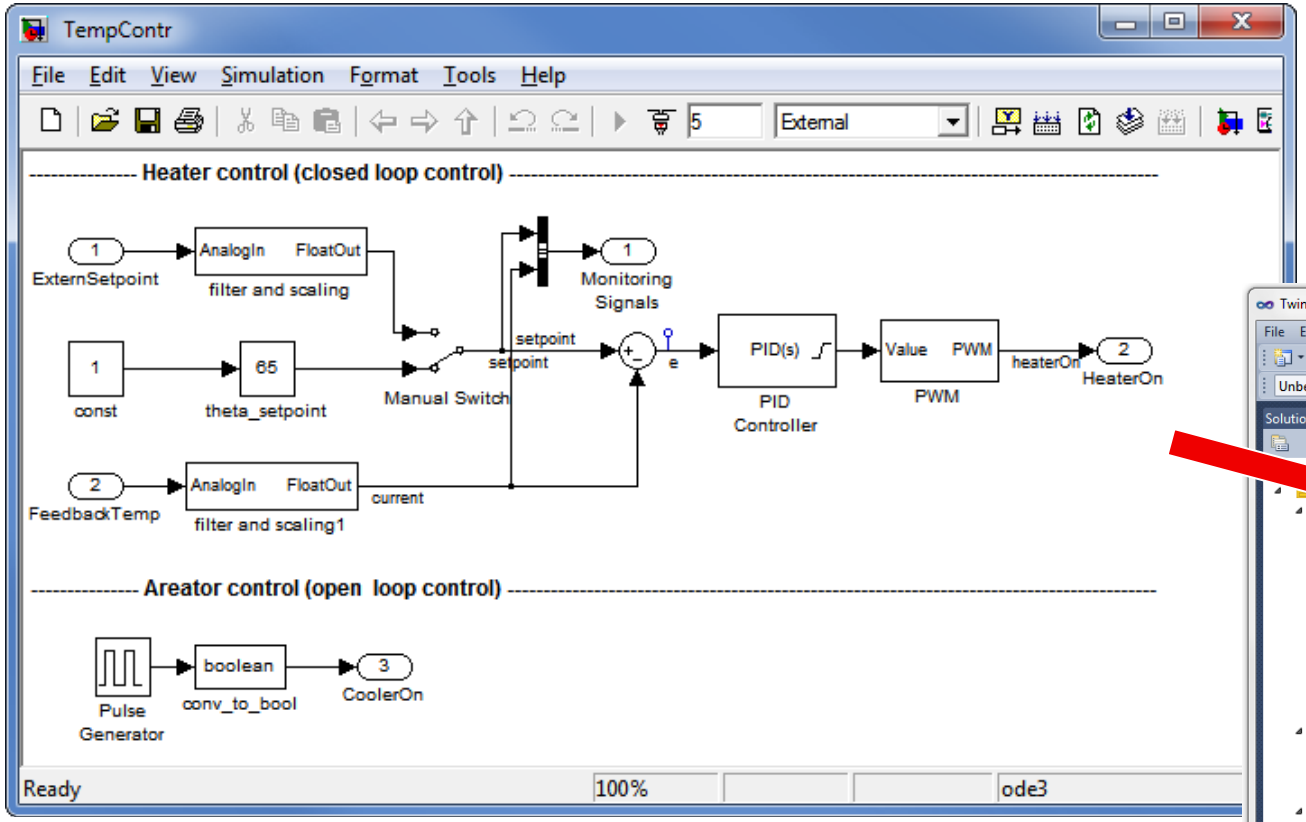
Partner

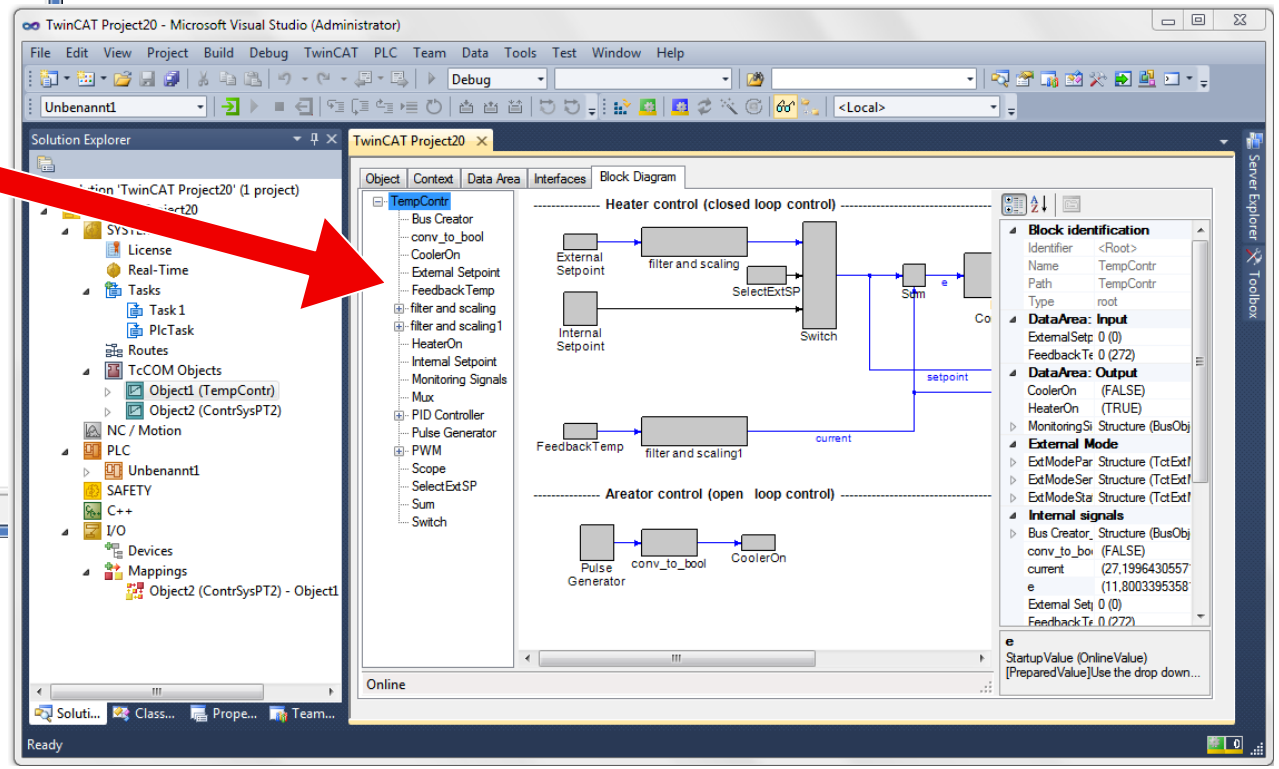
Code generation

- design in Simulink®
- automatic generation of C code by the Simulink Coder™
- compilation with Visual Studio® C Compiler
- parameterization in the TwinCAT System Manager
- download and execution in the TwinCAT 3 Runtime

MATLAB ®/Simulink® integration

BECKHOFF





C# / .Net

.NET Programming Languages (e.g. C#)

- well-known programming languages
- standardized C# (ISO/IEC 23270)
- creates intermediate code (Common Intermediate Language – CIL)

benefits:

- efficient engineering with higher abstraction level
- widely accepted
- “Garbage collection” takes care on memory
- could now be handled as part of one integrated solution

restrictions:

- Garbage collector is not suitable for real-time applications

C#/.NET programming

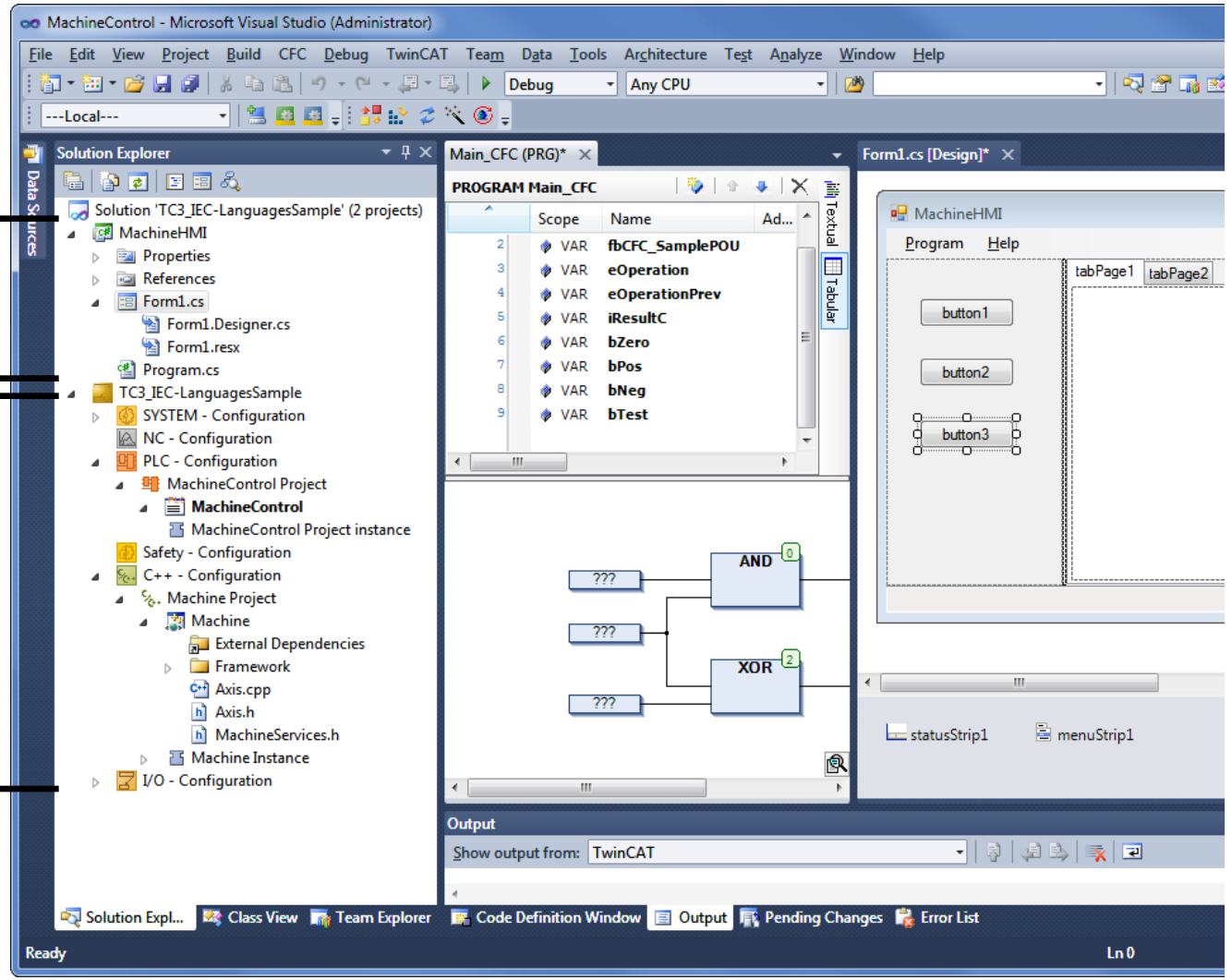
BECKHOFF

HMI/
Windows processes

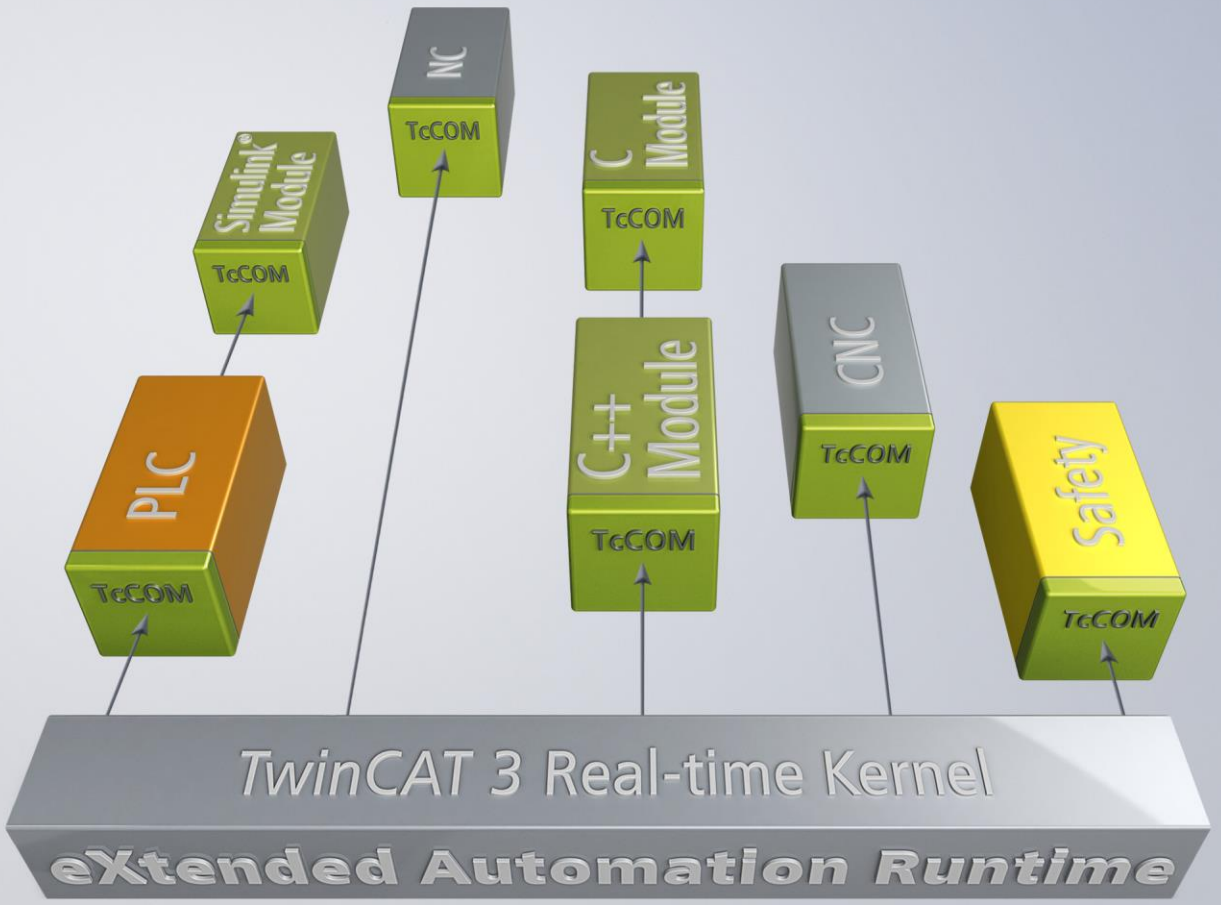
PLC module

C++ module

Real-time Code



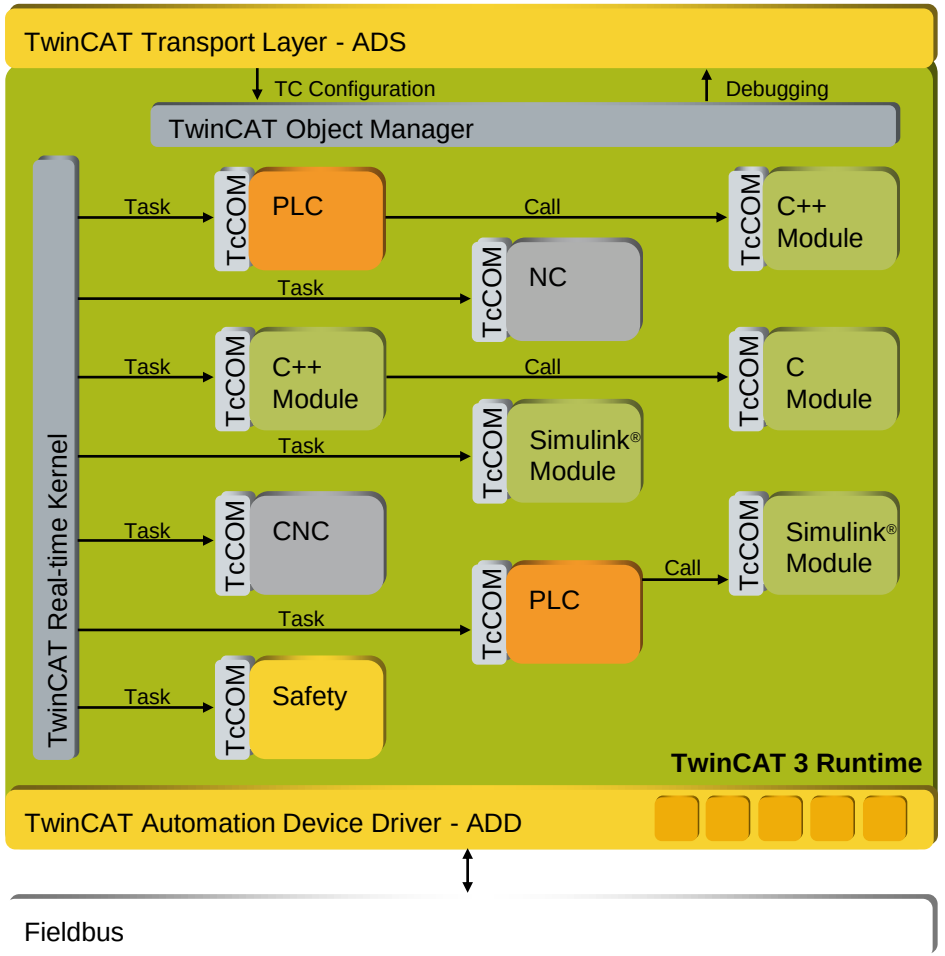
Runtime (XAR)


NC
TcCOM
Simulink® Module
TcCOM
C Module
TcCOM
PLC
TcCOM
C++ Module
TcCOM
CNC
TcCOM
Safety
TcCOM
TwinCAT 3 Real-time Kernel
eXtended Automation Runtime

Modular runtime

BECKHOFF

- **dynamic** environment for the execution and administration of TwinCAT 3 modules
- administration of runtime modules (via TwinCAT Object Manager)
- defined interfaces (e.g. TwinCAT Component Object Model – TcCOM) and v model



Modular runtime interface

BECKHOFF

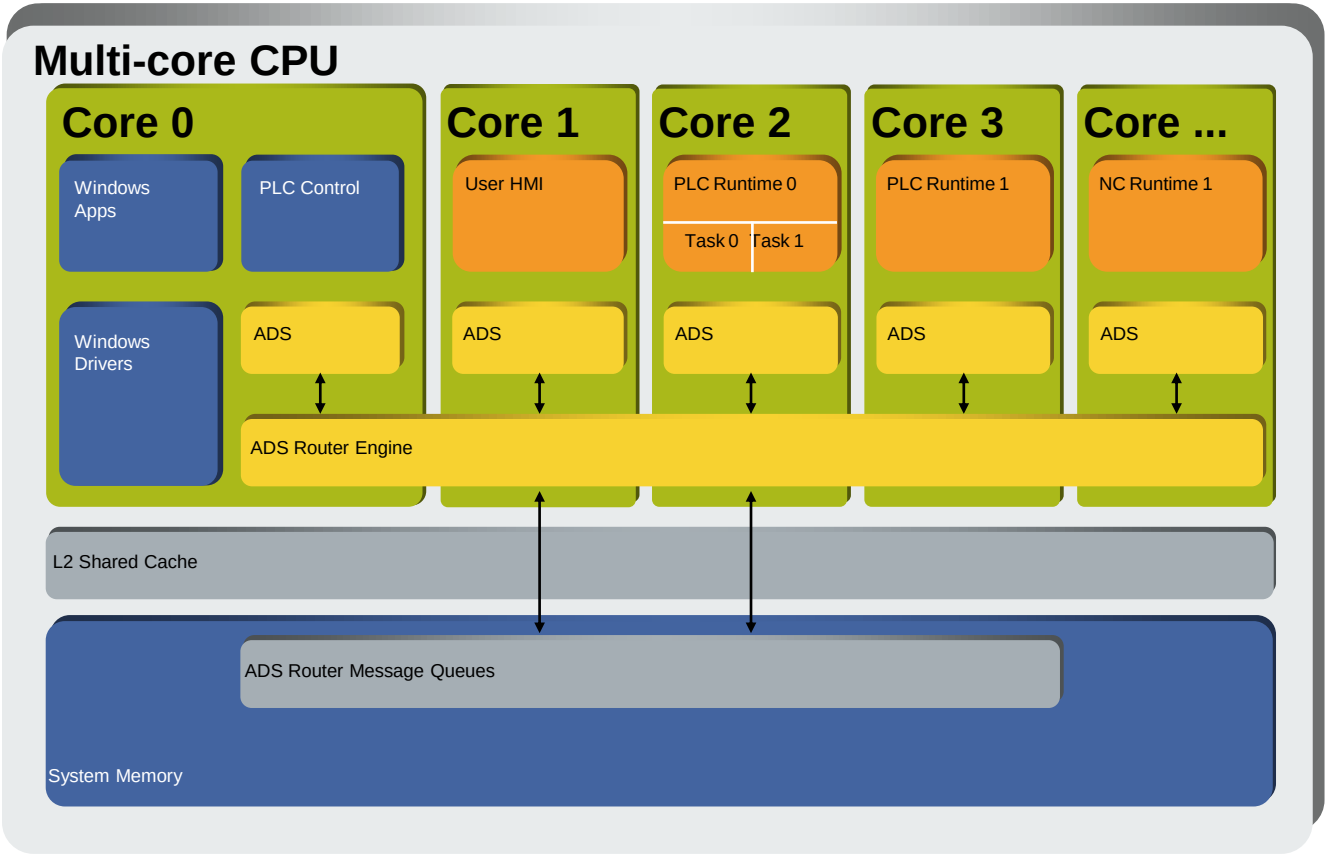
- separation of encapsulated functionality into modules
- extension of the base system by own drivers (Automation Device Drivers – ADD) e.g. fieldbus drivers
- scalability: modules can contain simple functions, complex algorithms and real-time tasks or complete projects
- reusability of modules
- cooperation of modules written in
  - IEC 61131-3
  - C/C++
  - MATLAB®/Simulink - generated modules

Multicore support

Support of multi-core systems

BECKHOFF

- distribution of projects to cores
  - (e.g. PLC, NC, Motion Control and HMI run on different cores)
- scalable base time for each core
- scalable CPU usage for each core

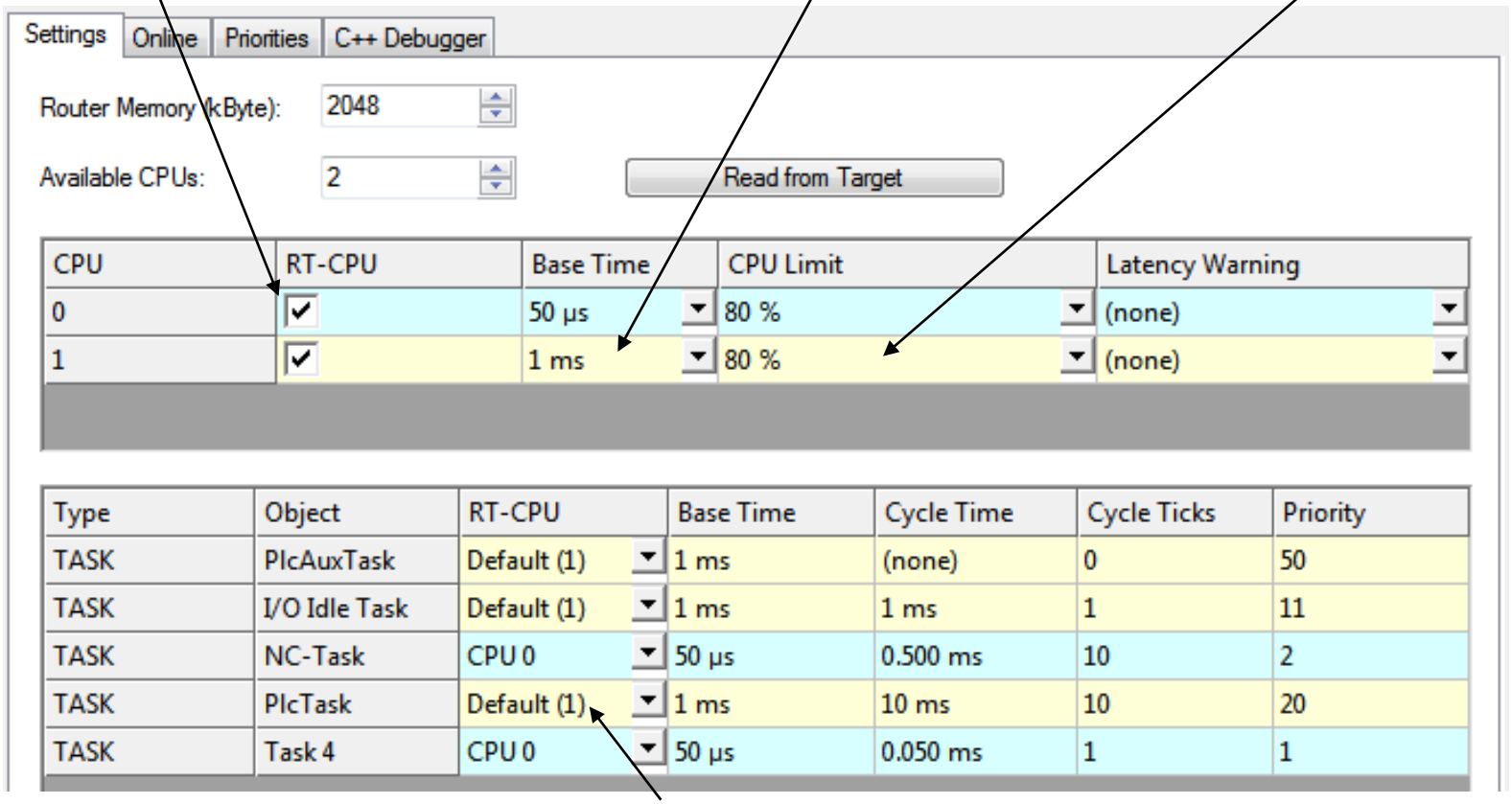


Support of multi-core systems

Enables cores for real-time usage

Defines the base time for a core

Defines the CPU-usage limit



Assignment of a task to a CPU

Benchmark of TwinCAT 3 Multicore

Testcase hardware:

- Intel® Core i7 950
- 4 physical core CPU at 3.07 GHz
- Nvidia 9800 graphic adapter

Testcase software:

- 4 PLC runtime systems
- each PLC runtime mapped to one CPU core
- all PLC runtimes execute identical benchmark software

Execution time for 1000 PLC commands (µs)

	Core0	Core1	Core2	Core3
bool	0.887	0.894	0.898	0.895
byte	0.672	0.682	0.681	0.688
word	0.613	0.614	0.626	0.617
dword	0.575	0.583	0.583	0.583
sint	3.463	3.472	3.473	3.474
int	3.473	3.484	3.482	3.482
dint	3.487	3.497	3.491	3.496
real	1.813	1.822	1.818	1.820
lreal	4.761	4.769	4.770	4.769

Result:

- TwinCAT Multicore technology increases PLC power, linear related to the number of CPU cores
- PC Control enables future CPU development for automation applications

Contents

BECKHOFF

Object-orientation by the IEC 61131-3 3rd edition

Benefits of the object-oriented extensions

- increased readability of the code by encapsulation algorithms into methods
  → improved maintainability
- modularization, structuring of the code
  → improved reusability
- abstract programming by using interfaces
  → improved extensibility and adaptability
- construction of inheritance hierarchies
  → improved extensibility and adaptability

Consistent usage of the object-oriented extensions enables:

- improved software quality
- decreased time for programming and maintenance

Object-orientation by the IEC 61131-3 3rd edition

BECKHOFF

Language features	2nd edition IEC 61131-3	3rd edition IEC 61131-3	C++	Java	C#
Multi-language support	+	+	-	-	-
Mixed programming OO/proc.	-	+	+	-	-
Classes	~ (FB)	+	+	+	+
Methods	~ (actions)	+	+	+	+
Interfaces	-	+	-	+	+
Partial abstract classes	-	-	+	+	+
Polymorphism	-	+	+/-	+	+
Reference semantics	-	+ (interfaces)	-	+	+
Constructor/destructor	-	+	+	+	+
Properties	-	+	-	-	+
Visibility	~ (variables)	~ (variables)	+	+	+
Dyn. memory (“new”)	-	- (in TwinCAT 3)	+	+	+

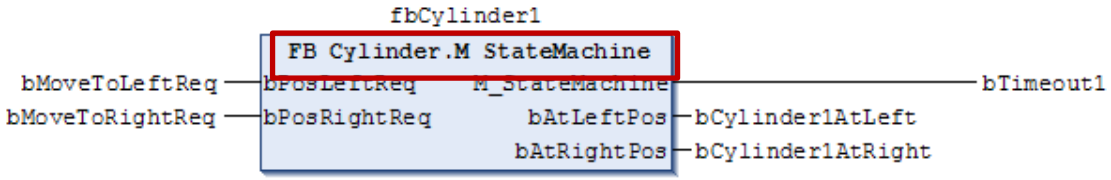
What does Object Orientation look like in TwinCAT 3?

BECKHOFF

- declaration of a function block

```
FUNCTION_BLOCK AC_Meldung EXTENDS TWINCAT_MODULE_BASE_LIBRARY.BaseModule IMPLEMENTS IAC_MELDUNG, ITCCHILD
```

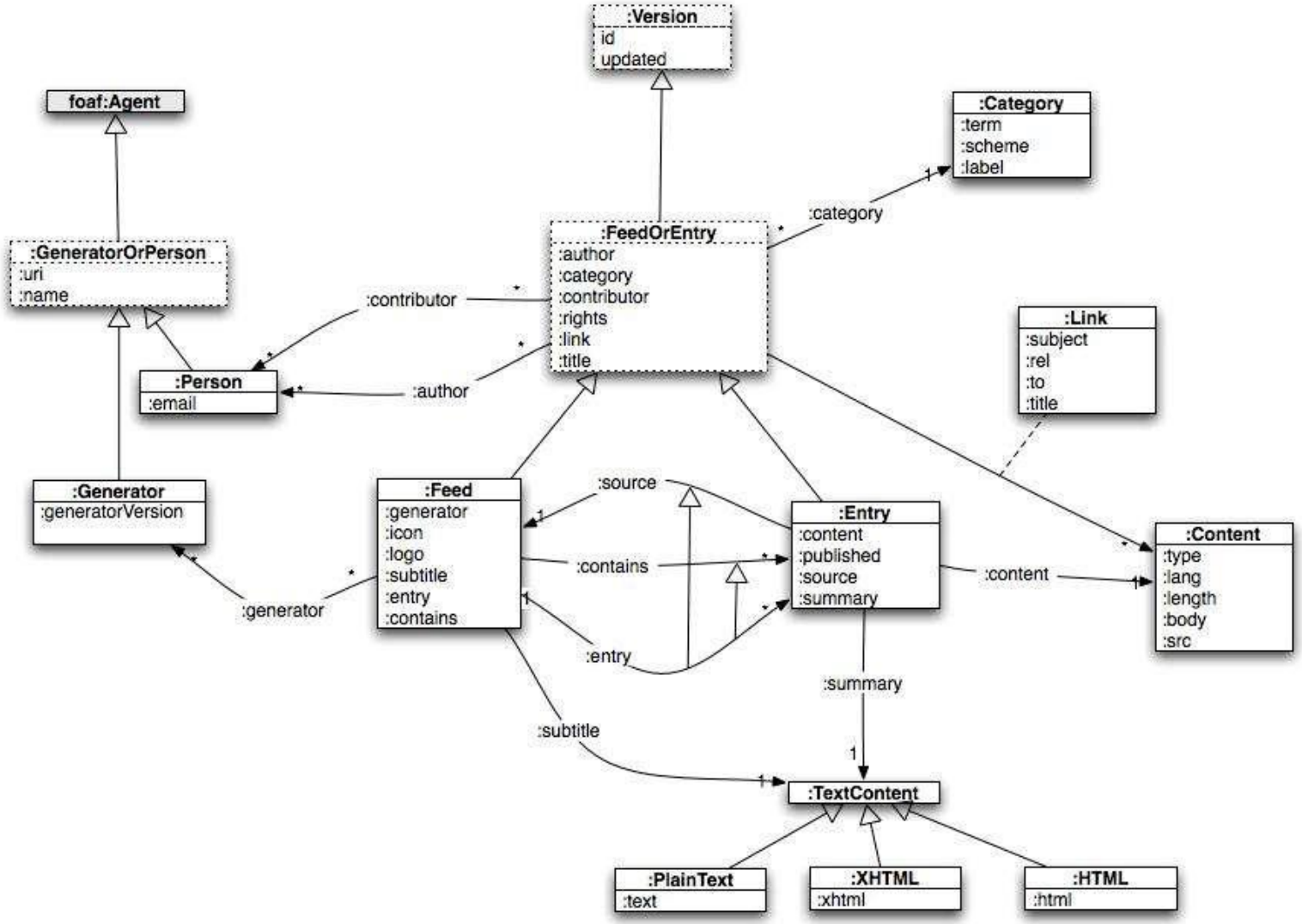
- call of a method



Usage of the object-oriented extensions:

- is possible in all IEC languages
- is independent from the used hardware
- is not mandatory!

UML – Unified Modeling Language

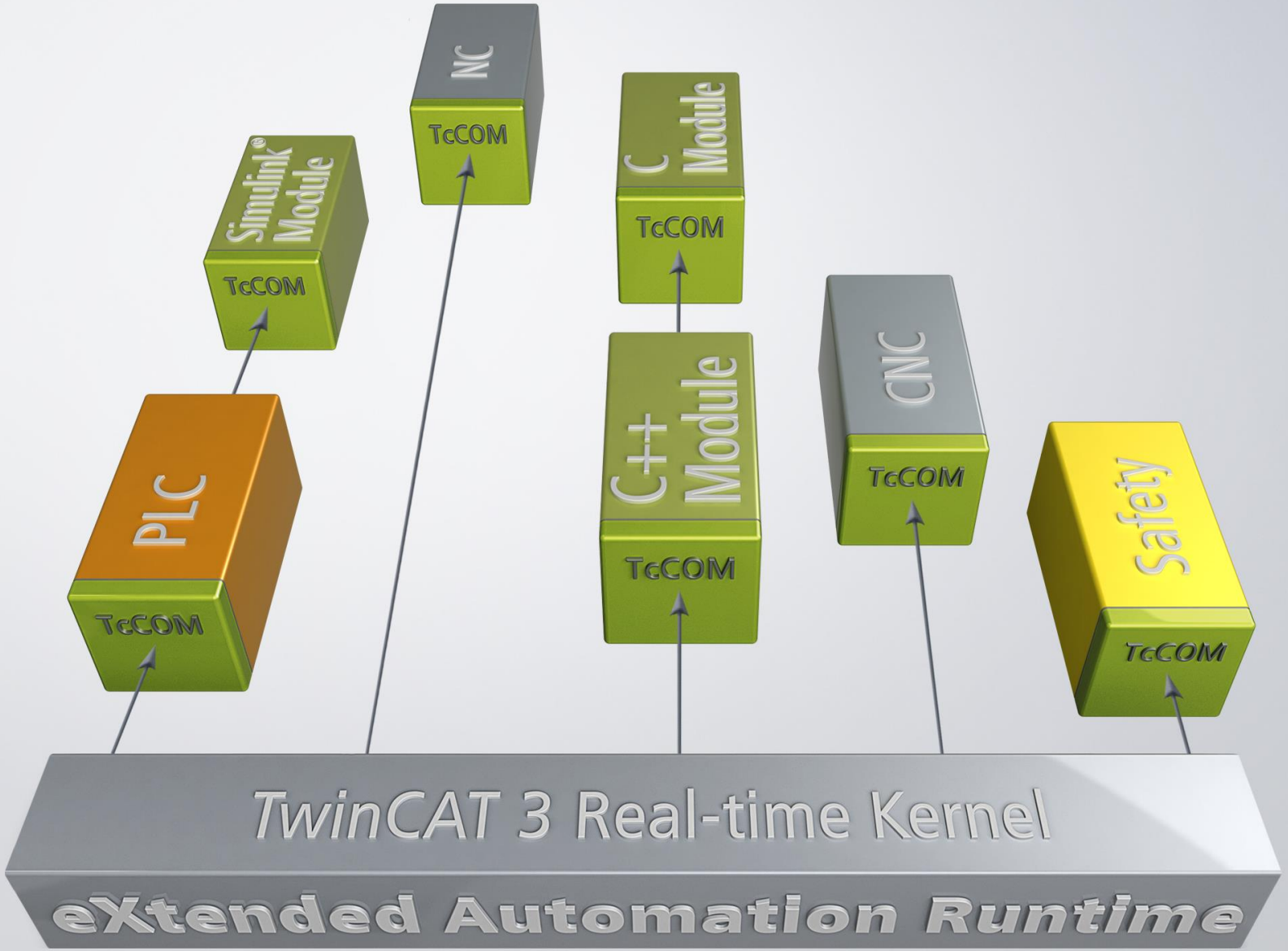


Contents

BECKHOFF

TcCOM – TwinCAT Component Object Model

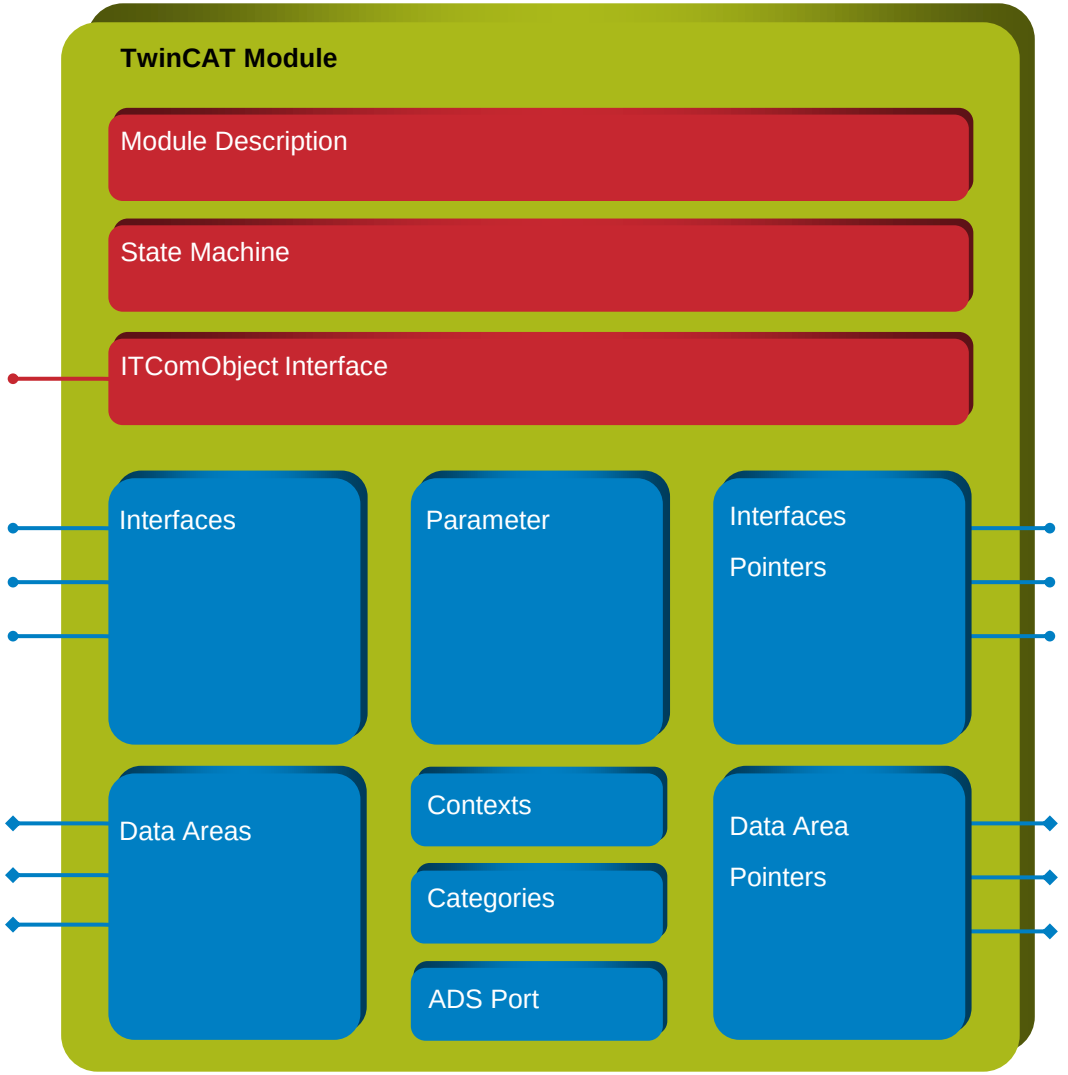
BECKHOFF



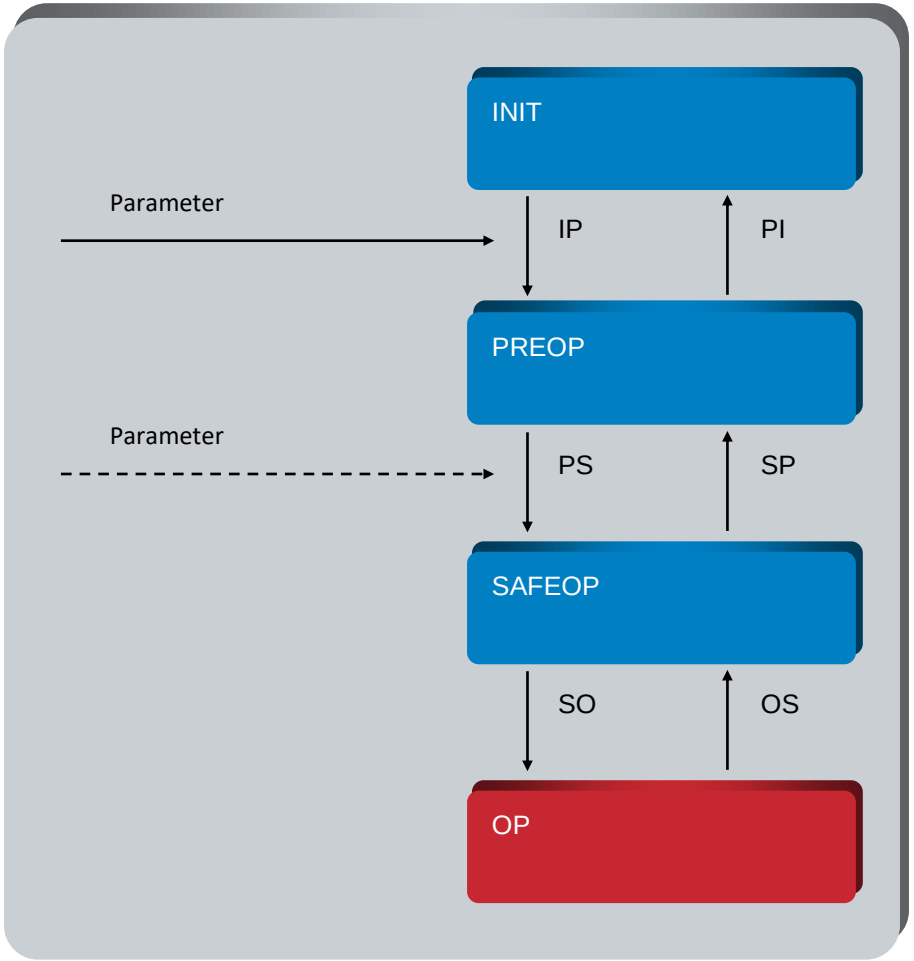
Modular runtime interface for modules

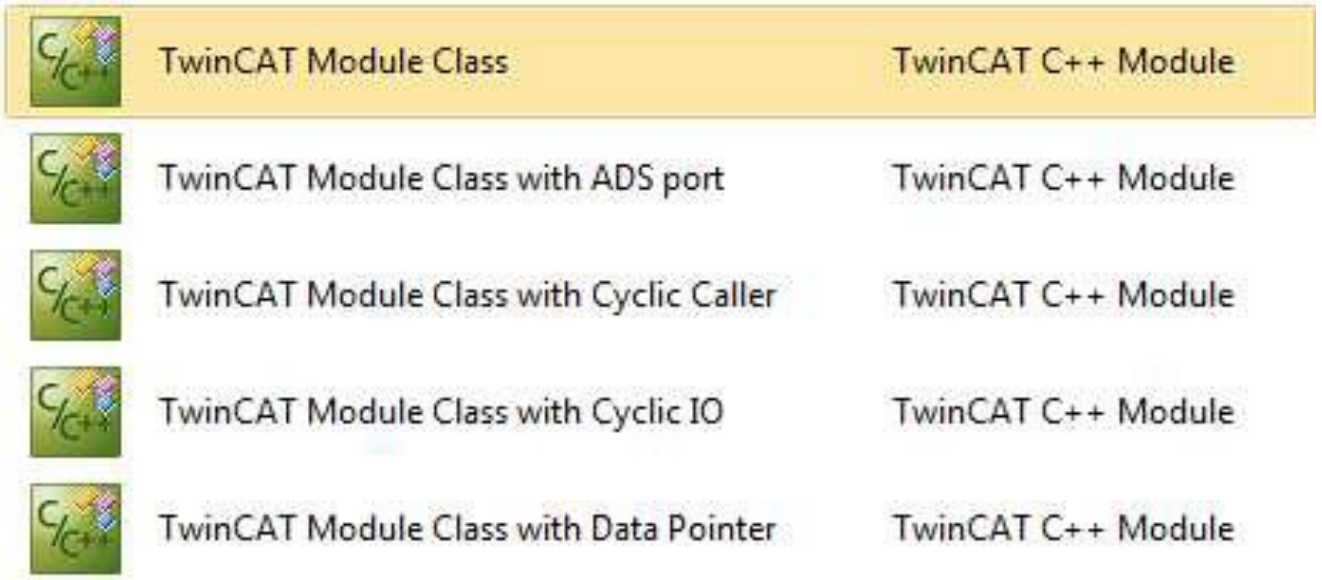
BECKHOFF

- standardised
- easy to use
- included state machine



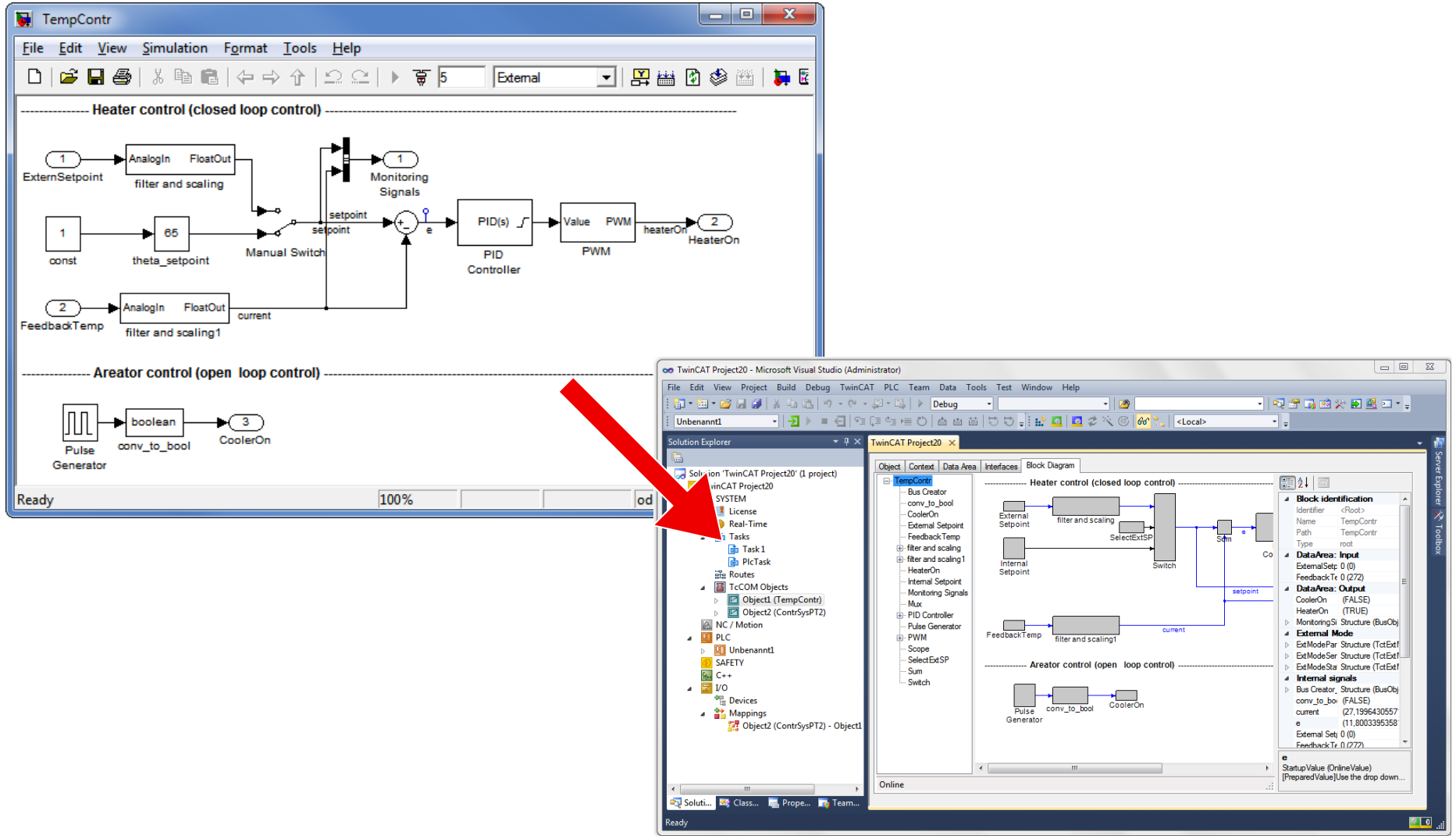
TwinCAT Module State Machine




TwinCAT Module Class
TwinCAT C++ Module
TwinCAT Module Class with ADS port
TwinCAT C++ Module
TwinCAT Module Class with Cyclic Caller
TwinCAT C++ Module
TwinCAT Module Class with Cyclic IO
TwinCAT C++ Module
TwinCAT Module Class with Data Pointer
TwinCAT C++ Module

MATLAB®/Simulink® integration

BECKHOFF



Contents

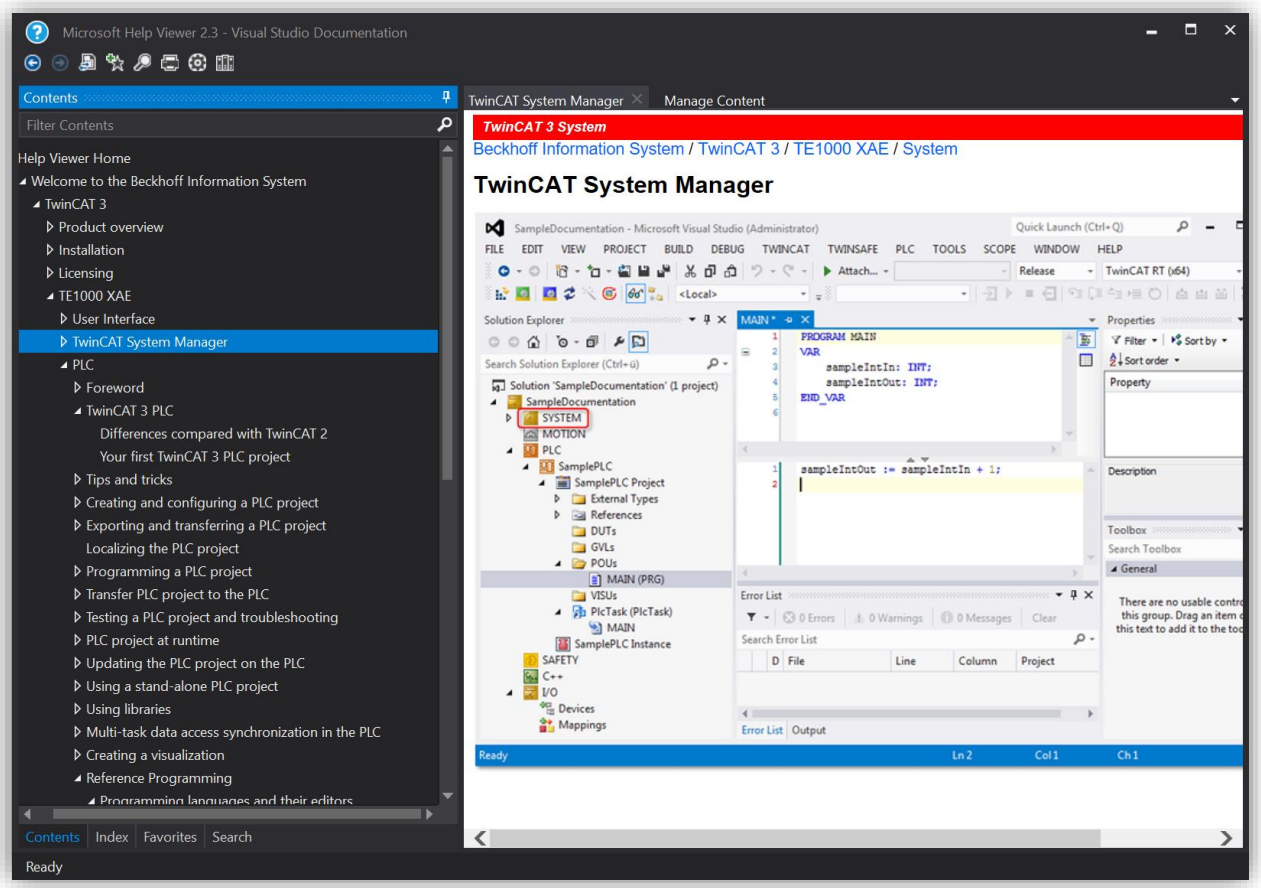
BECKHOFF

Documentation

BECKHOFF

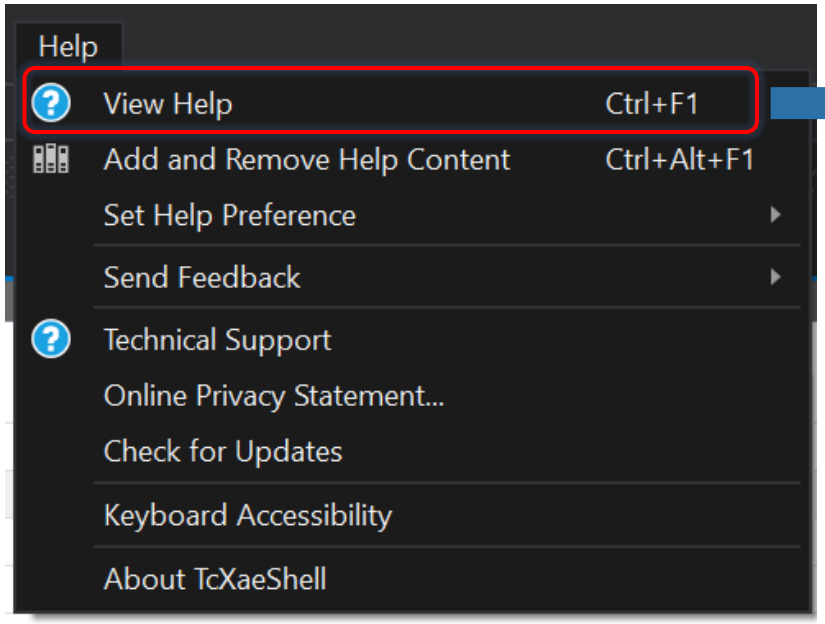
- Online –
  - https://infosys.beckhoff.com/

- Built in MS Help

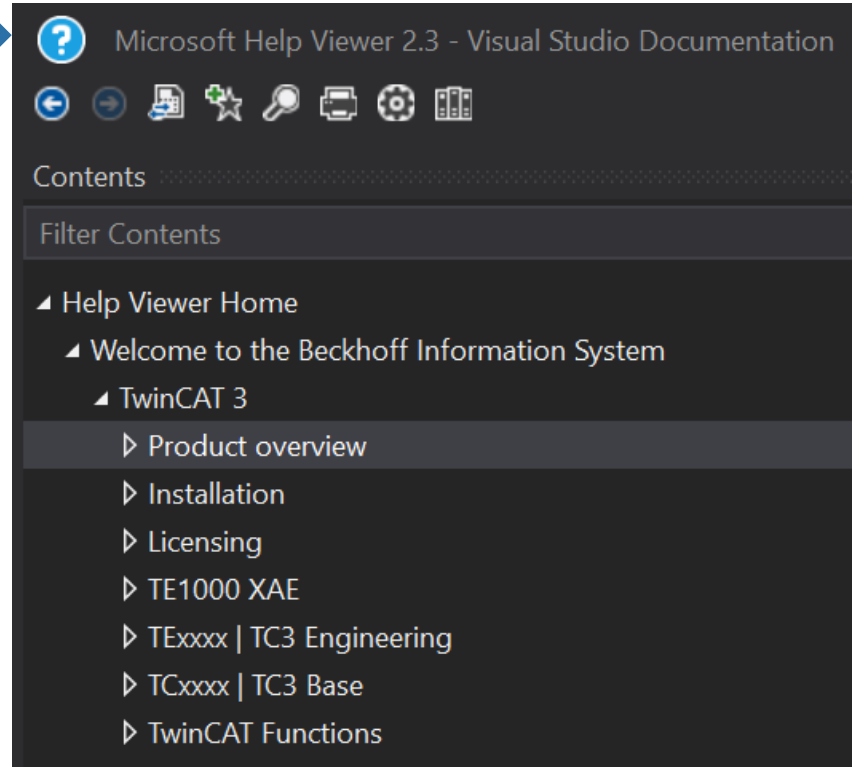


BECKHOFF

- MS Help integration*



Automatic Online Updates



* Additional installation required

For further information see www.beckhoff.com/TwinCAT3

Contact

BECKHOFF

Beckhoff Automation Ltd
Canadian Headquarters
4 Schiedel Ct, Unit 1-3
Cambridge
Ontario, N3C 0H1
Canada

Phone: +1 (289) 627-1900
E-mail: info@beckhoff.ca
Web: www.beckhoff.ca