C/C++
MATLAB®/
SIMULINK®

TwinCAT 3.1 delivers exciting new features for eXtended Automation

The integration of TwinCAT into Visual Studio® connects the realm of proven Beckhoff automation technologies with Microsoft's standard development environment for software. In this way, two worlds grow together: IT and automation technology programmed using one Visual Studio® platform, running on one PC, with different programming languages, all with highly deterministic execution of projects on a multi-core PC. The integration of TwinCAT into Visual Studio® offers the optional possibility to use source code control procedures familiar to IT experts for automation technology as well. Automatic code generation with the TwinCAT automation interface has been significantly extended once again in TwinCAT 3.1. In addition, the TwinCAT 3.1 runtime can reserve individual cores of multi-core processors exclusively for the execution of TwinCAT tasks.

Author: Dr. Josef Papenfort,
TwinCAT Product Manager, Beckhoff

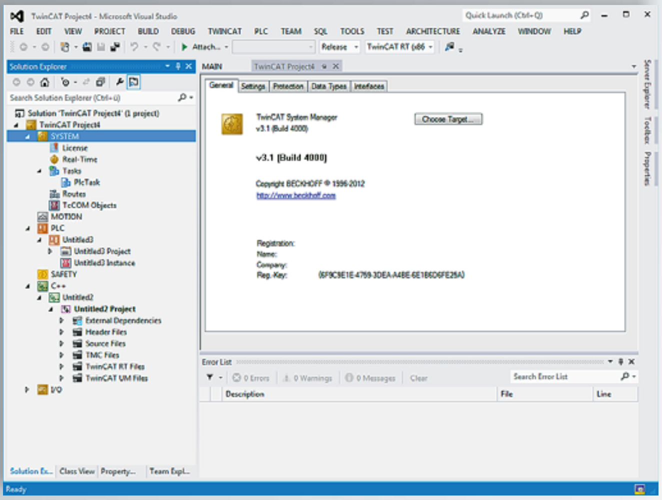
Support for the new Visual Studio® 2012

TwinCAT 3 was introduced to the market with Visual Studio® 2010; version 3.1 can also be used in the current release of Visual Studio® 2012. TwinCAT integrates itself automatically into the existing Visual Studio®. If none exists, the free Visual Studio® shell is installed and the TwinCAT 3 components are integrated into this shell. If several Visual Studio® versions exist on a PC, TwinCAT integrates itself into all existing and supported versions.

Improved support for source code control tools

Source code control tools can be used for different application cases. Apart from saving and backing up source code, their main use is no doubt for revision control. Each programmer who, for example, modifies a PLC function block, must "check this in" in the source code administration. The change is thus permanently saved. It is known at all times who created or modified each version of this function block as well as when and why. Appropriate comparison tools are available in TwinCAT 3.1 for displaying the differences between the versions. Initially the differences can be displayed in the project tree, the so-called Solution Tree. It is possible to view here at a glance which nodes have been modified. If a difference is detected, further details can be made visible by a double click. The comparisons are available in languages that correspond to the different PLC programming languages.

In order to effectively use source code control tools, all programs and all configurations must be available in a readable form – i.e. not as binary files. In the TwinCAT 3.1 system, therefore, programs such as PLC function blocks and configurations are kept in XML files that can be saved easily in the source code database. The PLCopen XML format was relied upon for the PLC function blocks, data structures and task configurations. In comparison with the previous ASCII format, this results in a significant advantage. Whereas standardized export formats exist only for ST, IL and SFC in the IEC 61131-3 standard, XML files can be exported and imported for all languages and for the complete PLC configuration in PLCopen XML. TwinCAT 3 uses this standard. Apart from the

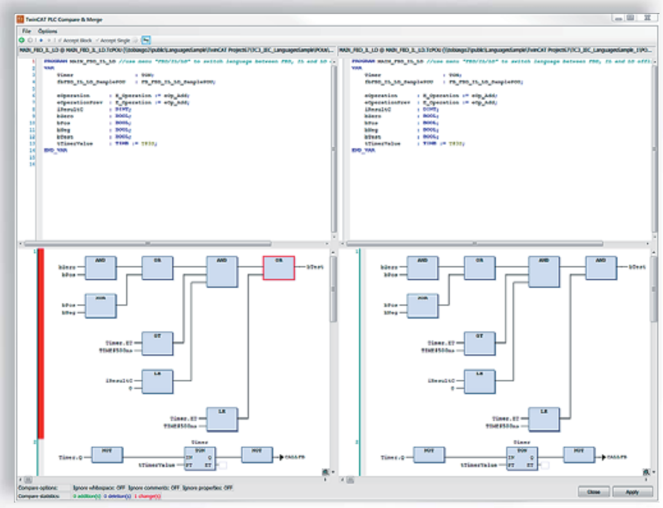

TwinCAT Project4 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TWINCAT PLC TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Solution Explorer
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
General Settings Protection Data Types Interfaces
Choose Target...
Error List

TwinCAT System Manager
v3.1 (Build 4000)

v3.1 [Build 4000]

Copyright BECKHOFF © 1996-2012
http://www.beckhoff.com

TwinCAT 3.1 integrates into Visual Studio® 2012



Comparison of various versions of function blocks that are saved in the Team Foundation Server

PLC data, all Motion Control and fieldbus configuration data are also saved in XML. Naturally, C and C++ files continue to be saved in the native ASCII format.

In principle, a whole series of plug-ins for various source code control tools are available for Visual Studio®. The most important are Microsoft Team Foundation Server (TFS) and Subversion. Both are fully supported by TwinCAT.

Automatic code generation with the TwinCAT Automation Interface

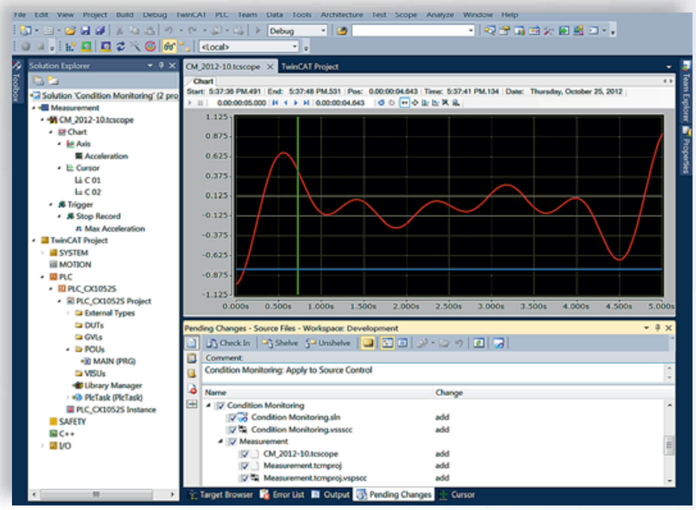
It is clear that engineering costs are increasing considerably. Manual work and the errors associated with it can be reduced by complete or partial automatic code generation. One example of this is the possibility to accept configuration data from an electro-CAD. After all, the controllers, axes and I/Os are already created in electro-CAD. These configuration data can be re-used automatically in a TwinCAT configuration.

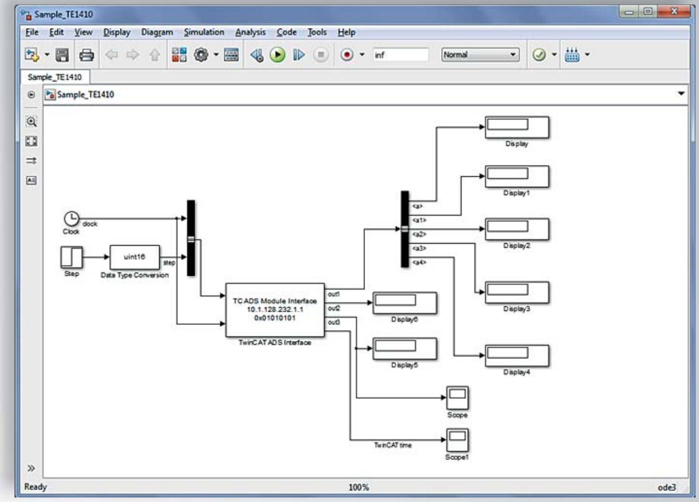
In TwinCAT these automated configuration settings are made via the TwinCAT Automation Interface. This interface, which is already familiar from TwinCAT 2 and is frequently used, has been greatly expanded in TwinCAT 3.1. For instance, it is now possible to generate and compile PLC and C++ code automatically. Matlab®/Simulink® modules can be added automatically to a project and configured. All of this is possible without human intervention.

The Automation Interface can be used by different script languages such as IronPhyton and Powershell and linked into a C++ or .NET application. Consequently, the Automation Interface can also deal with a source code control tool.

Support for different platforms

The TwinCAT system is conceived in such a way that different automation platforms are supported: I/O, PLC and Motion Control are executable on different processors and on different operating systems without the programmer having to adapt the software. The problem is solved by having different code generators for the various platforms. TwinCAT version 3.1 supports the runtimes for 64-bit operating systems and for CE platforms with x86 processors. TwinCAT can also be operated with the engineering tools and runtime on the new Microsoft Windows 8 operating system. What is new in TwinCAT 3.1 is the possibility to reserve individual cores of a multi-core processor exclusively for TwinCAT so that they are no longer available to the "normal" Windows operating system. This


Integration of Scope Configuration and View



Matlab®/Simulink® connector

feature, known as "CPU Isolation," gives the full performance of the reserved cores exclusively to TwinCAT.

Integration of further tools into the Visual Studio® Framework

A consistent step toward simplifying configuration is the integration of further configurators into Visual Studio®. These include the tried-and-tested TwinCAT Scope. The scope configuration is a self-contained project and thus part of the solution. Using it, configuration data can be simply embedded into the source code administration. Variables can be added directly to channels of a scope by drag and drop. It is also possible to display the variables in the monitoring window of the PLC and in parallel as a graph on the scope. This simplifies debugging and reduces engineering costs.

Functions extend the functionality of TwinCAT 3.1

With the introduction of TwinCAT 3.1, Scope is also available in a professional version. Here, a configuration and display tool as well as an extended server are available for extended functionalities in addition to the tools provided in the free basic version. Long-term recordings, improved triggering and the possibility to use the scope for .NET control in one's own applications are all achievable.

Where support for Matlab®/Simulink® is concerned, a Simulink® block is available in the "TwinCAT interface for Matlab®/Simulink®" that establishes an ADS connection to real-time variables. This allows direct access from Simulink® to variables, for example from the I/O area or the PLC, via a convenient configuration. In hardware-in-the-loop, this means variables from the fieldbus can be transported directly into Simulink® and used further. If, for example, variables of a PLC are directly exchanged with variables of a simulation in Simulink®, then this is called "software-in-the-loop."

Further Information:
www.beckhoff.com/TwinCAT3