
C/C++
MATLAB®/
SIMULINK®

TwinCAT 3.1 – Weitere neue Eigenschaften für eXtended Automation

Autor: Dr. Josef Papenfort,
Produktmanager TwinCAT, Beckhoff

Die Integration von TwinCAT in das Visual Studio® verbindet die bewährte Beckhoff-Automatisierungstechnik mit der Standard-Entwicklungsumgebung für Software von Microsoft. Damit wachsen Welten zusammen: IT und Automatisierungstechnik in einem Visual Studio®, auf einem PC, mit unterschiedlichen Programmiersprachen, hochdeterministisch auf einem Multicore-PC ausgeführt. Die Integration von TwinCAT in das Visual Studio® bietet die Möglichkeit, auch die aus der IT bekannten Verfahren zur Source-Code-Kontrolle für die Automatisierungstechnik zu nutzen. Automatische Codegenerierung mit dem TwinCAT Automation Interface wurde in TwinCAT 3.1 noch einmal stark erweitert. Außerdem kann die Runtime von TwinCAT 3.1 auf Mehrkernprozessoren einzelne Kerne exklusiv für die Ausführung von TwinCAT-Tasks reservieren.

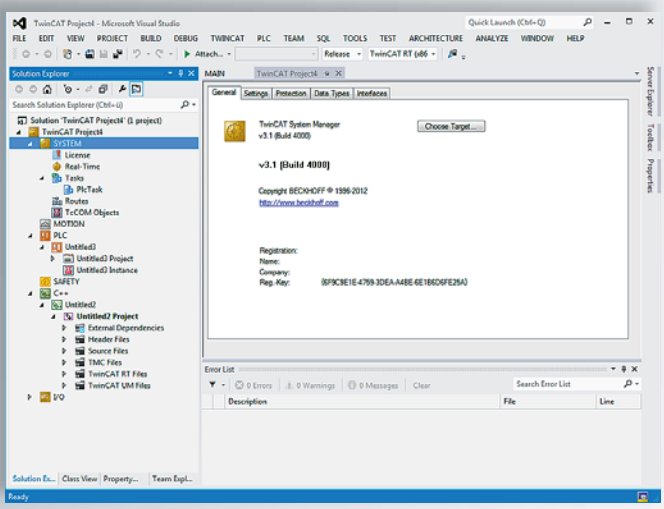
Unterstützung des neuen Visual Studio® 2012

TwinCAT 3 wurde mit dem Visual Studio® 2010 in den Markt eingeführt; die Version 3.1 kann auch im aktuellen Visual Studio® 2012 betrieben werden. TwinCAT integriert sich automatisch in das vorhandene Visual Studio®. Ist keines vorhanden, wird die kostenlose Visual Studio® Shell installiert und die TwinCAT-3-Komponenten werden in diese Shell integriert. Sollten mehrere Visual-Studio®-Versionen auf einem PC vorhanden sein, integriert TwinCAT sich in alle vorhandenen und unterstützten Versionen.

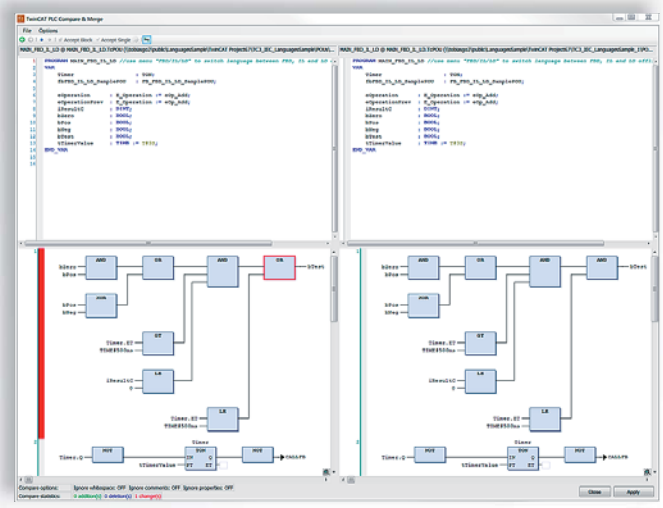
Verbesserte Unterstützung von Sourcecode-Controltools

Sourcecode-Controltools können für verschiedene Anwendungsfälle genutzt werden. Schwerpunkt ist sicherlich – neben der Speicherung und Sicherung von Sourcecode – die Versionierung. Jeder Programmierer, der z. B. einen SPS-Baustein ändert, muss diesen in der Sourcecode-Verwaltung ‚einchecken'. Damit wird diese Änderung dauerhaft gespeichert. Zu jeder Zeit ist bekannt, welche Versionen dieses einen Bausteins von wem, wann und warum erzeugt oder geändert wurde. Zur Anzeige der Versionsunterschiede stehen in TwinCAT 3.1. entsprechende Vergleichswerkzeuge zur Verfügung. Zunächst können die Unterschiede im Projektbaum – dem sogenannten Solution Tree – angezeigt werden. Hier ist auf einen Blick erkennbar, welche Knoten geändert worden sind. Ist ein Unterschied festgestellt, können durch einen Doppelklick weitere Details sichtbar gemacht werden. Für die verschiedenen SPS-Programmiersprachen sind die Vergleiche in den entsprechenden Sprachen verfügbar.

Um Sourcecode-Controltools sinnvoll zu nutzen, müssen alle Programme und alle Konfigurationen in lesbarer Form – also nicht als binäre Dateien – vorliegen. Im TwinCAT-3.1-System werden deshalb Programme, wie z. B. SPS-Bausteine und Konfigurationen in XML-Dateien gehalten, die einfach in der Sourcecode-Datenbank gespeichert werden können. Für die SPS-Bausteine, Datenstrukturen und Taskkonfigurationen wurde auf das PLCopen-XML-Format zurückgegriffen. Gegenüber dem bisherigen ASCII-Format ergibt sich hier ein wesentlicher Vorteil. Während in der Norm IEC 61131-3 nur für ST, AWL und SFC standardisierte



TwinCAT 3.1 integriert in das Visual Studio® 2012



Vergleich von verschiedenen Bausteinversionen, die im Team Foundation Server gespeichert werden

Exportformate vorliegen, können in PLCopen XML für alle Sprachen und für die gesamte SPS-Konfiguration XML-Dateien exportiert und importiert werden. TwinCAT 3 nutzt diesen Standard. Neben den SPS-Daten werden auch alle Motion-Control- und Feldbus-Konfigurationsdaten in XML gespeichert. C- und C++-Dateien werden natürlich weiterhin im nativen ASCII-Format gespeichert. Prinzipiell sind für das Visual Studio® eine ganze Reihe von Plug-ins für verschiedene Sourcecode-Controltools verfügbar. Die wichtigsten sind Microsoft Team Foundation Server (TFS) und Subversion. Beide werden von TwinCAT vollständig unterstützt.

Automatische Codegenerierung mit dem TwinCAT Automation Interface

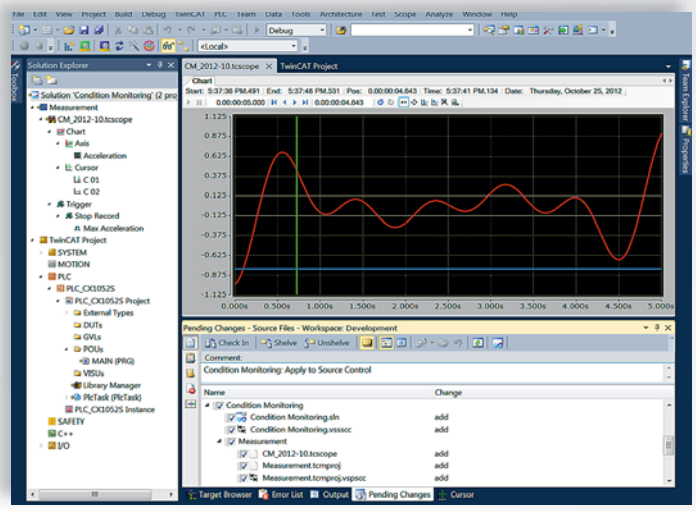
Engineeringkosten steigen erheblich. Durch vollständige oder teilweise automatische Codegenerierung können manuelle Arbeiten verringert und die damit verbundenen Fehler vermieden werden. Ein Beispiel ist die Möglichkeit, Konfigurationsdaten von einem Elektro-CAD zu übernehmen. Im Elektro-CAD werden bereits die Steuerungen, die Achsen und die I/Os angelegt. Diese Konfigurationsdaten lassen sich automatisch in einer TwinCAT-Konfiguration wiederverwenden.

In TwinCAT werden diese automatisierten Konfigurationseinstellungen via TwinCAT Automation Interface gemacht. Diese schon von TwinCAT 2 bekannte und oft verwendete Schnittstelle wurde in TwinCAT 3.1 stark erweitert. So ist es jetzt möglich, SPS- und C++-Code automatisch zu erzeugen und zu kompilieren. Matlab®/Simulink®-Module lassen sich automatisiert zu einem Projekt hinzufügen und konfigurieren. Und das alles ohne menschlichen Eingriff.

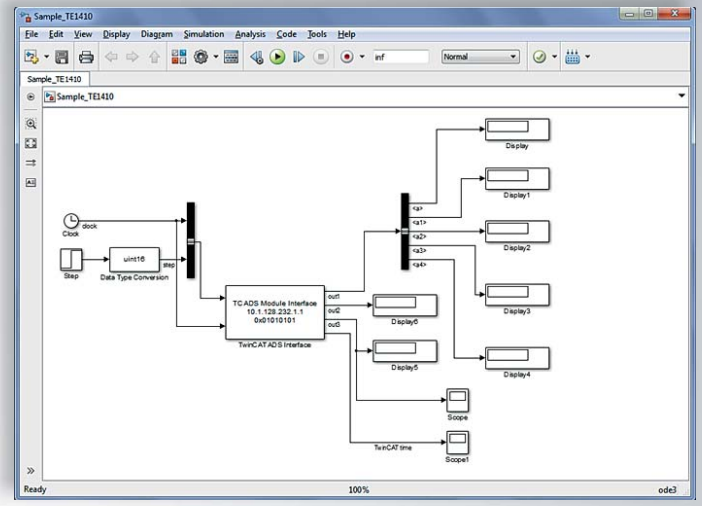
Das Automation Interface kann von verschiedenen Scriptsprachen wie IronPhyton und Powershell benutzt und auch in eine C++- oder .Net-Applikation eingebunden werden. Konsequenterweise kann das Automation Interface auch mit einem Sourcecode-Controltool umgehen.

Unterstützung verschiedener Plattformen

Das TwinCAT-System ist so konzipiert, dass unterschiedliche Plattformen unterstützt werden: I/O, SPS und Motion Control sind auf verschiedenen Prozessoren und auf verschiedenen Betriebssystemen ausführbar, ohne dass der Programmierer deswegen Anpassungen an der Software vornehmen muss. Das Problem wird durch unterschiedliche Code-Generatoren für die verschiedenen Plattformen gelöst. Die TwinCAT-Version 3.1 unterstützt die Runtimes für das 64-Bit-Betriebssystem und für die CE-Plattformen mit x86-Prozessoren. Auch unter dem neuen Microsoft-Betriebssystem Windows 8 kann TwinCAT mit den Engineeringwerkzeugen und der Runtime betrieben werden. Neu ist die Möglichkeit, mit TwinCAT 3.1 einzelne Kerne eines Mehrkern-



Integration von Scope-Konfiguration und View



Matlab®/Simulink®-Connector

prozessors exklusiv für TwinCAT zu reservieren, sodass sie für das ‚normale' Windows-Betriebssystem nicht mehr zur Verfügung stehen. Dieses „CPU-Isolation" genannte Feature stellt die reservierten Kerne in voller Performance ausschließlich für TwinCAT bereit.

Integration weiterer Tools in das Visual-Studio® Framework

Ein konsequenter Schritt in Richtung Konfigurationsvereinfachung ist die Integration weiterer Konfiguratoren in das Visual Studio®. Dazu gehört das schon lange bewährte TwinCAT Scope. Die Scope-Konfiguration ist ein eigenständiges Projekt und damit Teil der Solution. Mit ihr lassen sich Konfigurationsdaten einfach in die Sourcecodeverwaltung einbetten. Variablen können direkt – per Drag and Drop – zu Kanälen eines Scopes ergänzt werden. Möglich ist auch die Darstellung der Variablen im Monitoringfenster der SPS und parallel dazu noch als Graph im Scope. Das Debugging wird dadurch einfacher; die Engineeringkosten werden reduziert.

Functions erweitern den Funktionsumfang von TwinCAT 3.1

Das Scope ist mit der Einführung von TwinCAT 3.1 auch in einer Professional-Version verfügbar. Damit stehen, neben der kostenlosen Basisversion, für erweiterte Funktionalitäten ein Konfigurations- und Anzeigetool und ein erweiterter Server zur Verfügung. Langzeitaufnahmen, verbessertes Triggern sowie die Möglichkeit, das Scope als .Net-Control in eigenen Applikationen nutzen zu können, sind realisierbar.

Im Bereich der Matlab®/Simulink®-Unterstützung steht mit dem „TwinCAT-Interface for Matlab®/Simulink®" ein Simulink®-Block zur Verfügung, der eine ADS-Verbindung zu Echtzeitvariablen herstellt. Damit kann aus Simulink® heraus – über eine komfortable Konfiguration – direkt auf Variablen z. B. aus dem I/O-Bereich oder aus der SPS zugegriffen werden. So können, im Bereich Hardware-in-the-loop, Variablen vom Feldbus direkt in Simulin® transportiert und weitergenutzt werden. Wenn z. B. Variablen einer SPS direkt mit Variablen einer Simulation in Simulink ausgetauscht werden, spricht man von Software-in-the-loop.

weitere Infos unter:
www.beckhoff.de/TwinCAT3