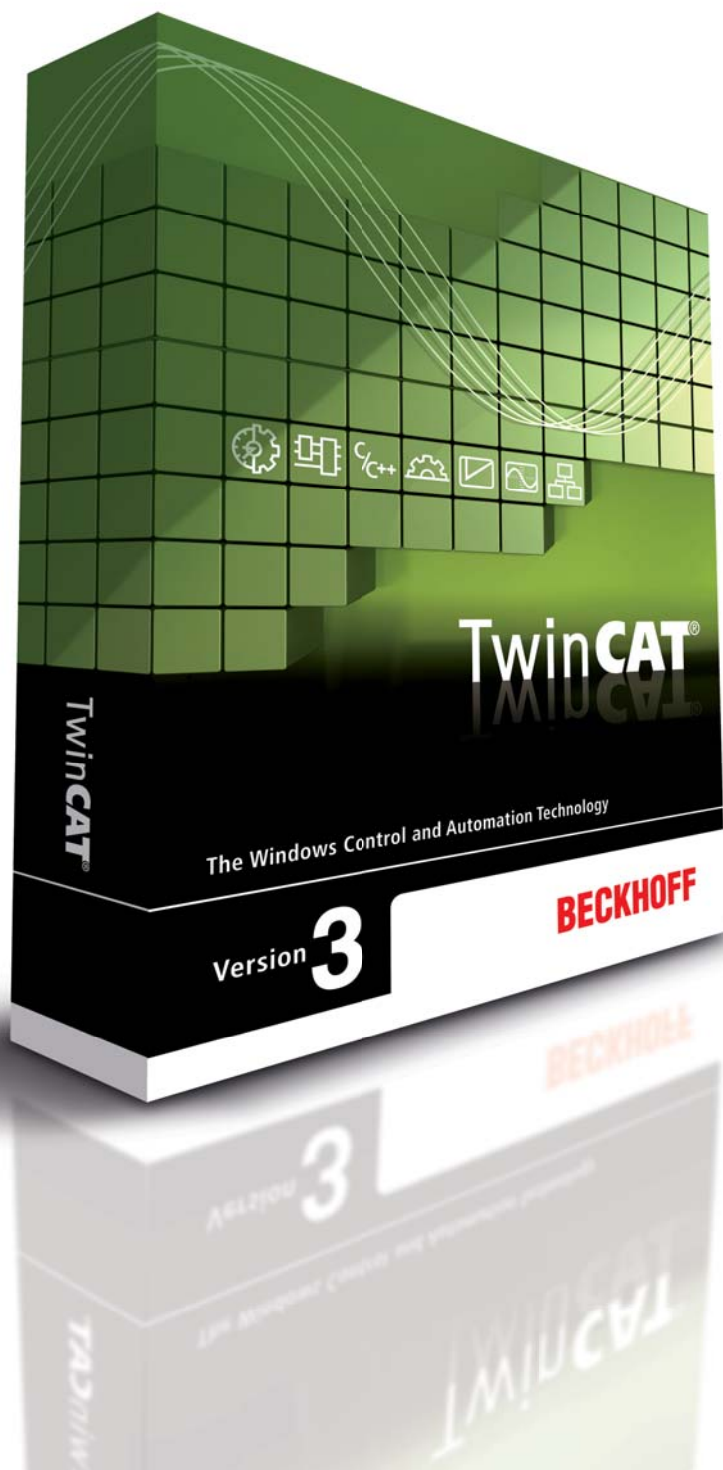


Interview zum Release von TwinCAT 3

Steuerungsprogrammierung nutzt Vorteile der SPS- und IT-Welt

Softwareprogrammierung geht heute über Kontaktplan hinaus. Dem muss auch der Softwareentwicklungsprozess Rechnung tragen: Mit der neuen Beckhoff-Softwaregeneration TwinCAT 3, die Standard-IT-Umgebung, Engineering mit Modulen und Runtime auf Multi-Core-Prozessoren umfasst, werden Steuerungsprogrammierer neue Wege beschreiten. Anlässlich des Software-Release von TwinCAT 3 zur SPS/IPC/DRIVES 2011 geben Ramon Barth und Dr. Dirk Jansen, als Leiter der Beckhoff-Software-Entwicklung und maßgeblich an der neuen Software-Architektur beteiligt, Auskunft über die Beta-Phase sowie die nächsten Punkte auf der Roadmap.



In der Automatisierungstechnik gewinnen integrierte Entwicklungsumgebungen immer mehr an Bedeutung. Zunehmend werden sie zum ausschlaggebendem Kriterium, aufgrund dessen Anwender ihre Systementscheidungen treffen. Die Erhöhung der Effizienz des Engineerings spielt dabei eine wichtige Rolle. TwinCAT, das Beckhoff-Softwaresystem, verwandelt nahezu jeden kompatiblen PC in eine Echtzeitsteuerung mit Multi-SPS-System, NC-Achsregelung, Programmierumgebung und Bedienstation. Es ermöglicht die Einbettung von IEC-61131-3-Software-SPS, Software-NC und Software-CNC in Windows-Betriebssysteme.

„Mit TwinCAT 3, der neuen Version unserer Automatisierungssoftware, haben wir eine Softwareplattform für die Automatisierungstechnik des nächsten Jahrzehnts geschaffen“, betont Ramon Barth, Leiter der Software-Entwicklung System, HMI und Echtzeit bei Beckhoff. „TwinCAT 3 erlaubt die optimale Nutzung moderner PC-Architekturen. Bei der Entwicklung stand nicht der Portalgedanke, sondern ein komplett offenes System mit einer vollständigen Produktpalette, auf der Agenda“, so der Softwarespezialist. Die neue Version bietet moderne SPS-Programmierung mit objektorientierten Erweiterungen, C++ und Matlab®/Simulink®-Integration und das Ganze in einer modularen Runtime, die von kleinen ARM-Chips bis hin zu modernen X86-Prozessoren mit Multi-Core genutzt werden.

TwinCAT 3 unterstützt Echtzeit-C++ auf Basis des marktführenden Compilers von Visual Studio®. „Die Integration der TwinCAT-3-Entwicklungswerkzeuge in eine Standardentwicklungsumgebung ermöglicht den Anwendern, auf alle Errungenschaften der IT-Welt zuzugreifen“, erklärt Ramon Barth. „Programme können modularer geschrieben werden. Hinzu kommen die Wiederverwendbarkeit von Code sowie die objektorientierten Mechanismen. Darüber hinaus laufen die Matlab®/Simulink®-Algorithmen direkt in der Echtzeit-Umgebung ab.“

Für SPS- und Hochsprachen-Programmierer geeignet

TwinCAT 3 ist, laut Ramon Barth, für zwei Anwendergruppen ideal ausgelegt: Zum einen sind dies die SPS-Programmierer, die sich in der Umgebung entsprechend der IEC-61131-3 zu Hause fühlen. Zum anderen zählen Anwender dazu, die mit C++ ihre Aufgaben lösen wollen. „Beides ist in TwinCAT 3 vorgesehen: objektorientierte Programmierung mit den Mechanismen der Vererbung sowie die Kontaktplanprogrammierung“, stellt Ramon Barth heraus.



Dr. Dirk Janssen, Leiter der Software-Entwicklung System, CNC und IO bei Beckhoff



Ramon Barth, Leiter der Software-Entwicklung System, HMI und Echtzeit bei Beckhoff

Die Einbindung in Visual Studio® ermöglicht es, einfach auf weitere Tools zuzugreifen. „Visual Studio® stellt alle modernen Hilfsmittel für die Programmierung zur Verfügung“, erläutert Dirk Janssen, Leiter der Software-Entwicklung System, CNC und IO. Hierzu zählt Ramon Barth zum Beispiel die UML-Integration und die Versionskontrolle. „Automatisierungsprojekte lassen sich nun so planen, wie es im IT-Bereich üblich ist, nämlich entsprechend dem Softwareentwicklungsprozess“, setzt er fort. Dies sei Stand der Technik und würde derzeit an den Hochschulen gelehrt. „Visual Studio® ist ein Standard, nach dem sich alle richten. Alternativen gibt es nur in der Open-Source-Welt, die aber nicht die Vielseitigkeit, die Verbreitung und die Performance mitbringt. Dirk Janssen ergänzt: „Wir stellen unseren Anwendern die gleichen Tools zur Verfügung, die wir selber für die Programmierung nutzen.“ So sei es nun auch möglich, dass Anwender das komplette Automatisierungsprojekt in Visual Studio® erstellen und es anschließend in TwinCAT 3 integrieren.

Wiederverwendbarkeit von Code

TwinCAT 3 bietet dem Software-Entwickler eine Infrastruktur, um wiederverwendbare Softwaremodule zu generieren. „Dies lebt die neue TwinCAT-Version durch die eigene Modularisierung vor“, sagt Ramon Barth. Das Prinzip beruht auf dem Austausch von Interfaces, deren Beschreibung und der Unterstützung von Techniken zur objektorientierten Programmierung (OOP). Diese Interfaces stellen strikte Verträge zwischen den Modulen dar und gewährleisten die Kompatibilität zu zukünftigen Entwicklungen. „TwinCAT liefert sozusagen den Rahmen und Teile der Implementation, in den der Anwender seine Module einbetten kann“, erklärt Ramon Barth und gibt zu bedenken, dass heutige Projekte oft von jungen Ingenieuren realisiert werden. „Diese wollen neue Ideen einbringen und fordern moderne Tools ein. Die jüngeren Programmierer sind mit den neuen Technologien vertraut und wollen diese auch nutzen. Zudem kann man auch langsam in diese neue Welt hineinwachsen, indem man beispielsweise zuerst nur einfache Eigenschaften, wie die Vererbung, nutzt.“ „Trotzdem sind auch die bisherigen SPS-Programmierer genügend berücksichtigt worden“, betont Dirk Janssen, „da die übliche IEC-61131-3-Programmierung in TwinCAT 3 weiter vorgesehen ist. Die objektorientierten Erweiterungen in der SPS sind nahtlos in die bestehende IEC-61131-3-Programmiersprache integriert, sodass kein Anwender gezwungen wird, neue Programmierparadigmen zu lernen

und einzusetzen.“ Darüber hinaus ist die Software skalierbar, sodass auch „abgespeckte“ Versionen zur Verfügung stehen.

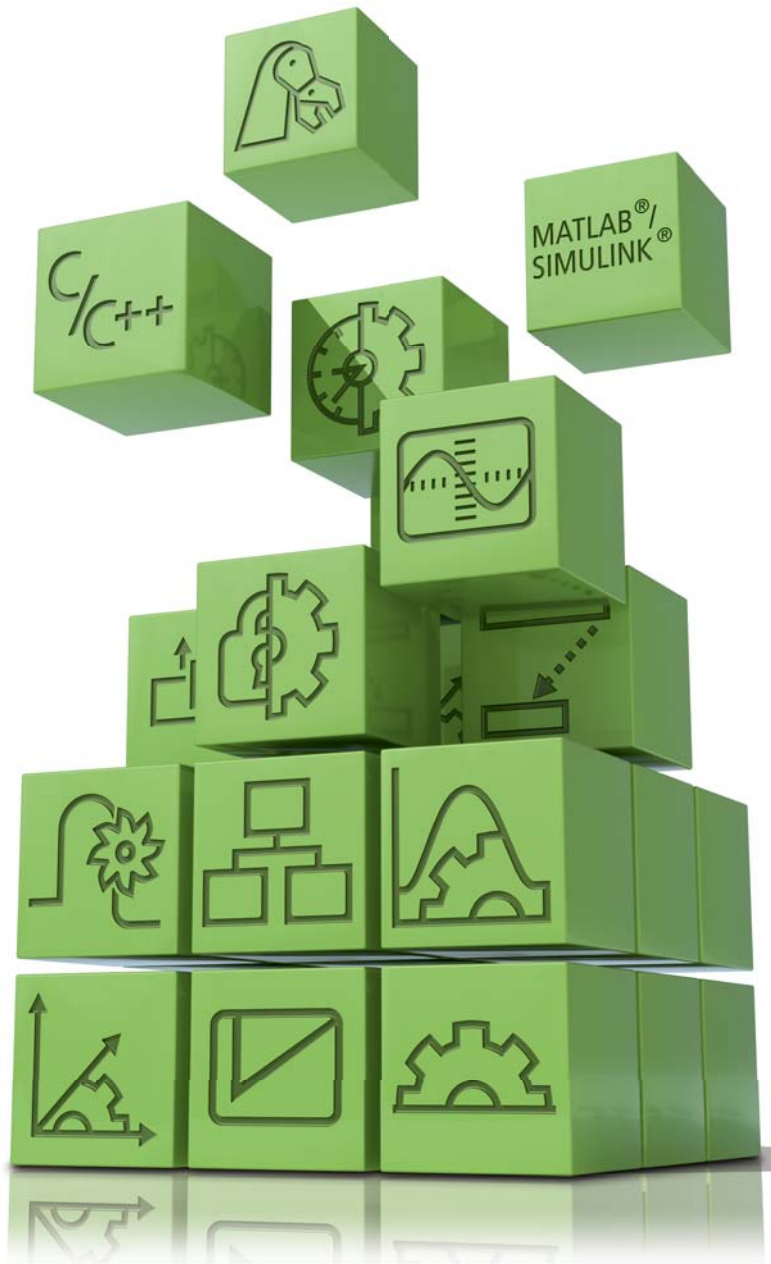
Dank der objektorientierten Programmierung lässt sich der Code leicht wiederverwenden, sodass sich viele Programmteile automatisch erstellen lassen“, unterstreicht Dirk Janssen. Dies war zwar auch schon in der Vorgängerversion möglich, jedoch unter Nutzung einer darüber liegenden Compiler-Schicht, was sich ungünstig bei nachträglichen Änderungen auswirkte. Ramon Barth gibt ein Beispiel für den praktischen Nutzen der Wiederverwendbarkeit von Code: „Ein Bohraggregat, das in verschiedenen Ausführungen eingesetzt wird, soll möglichst einheitlich von der Steuerung angesprochen werden. Daher definiert der Anwender die gemeinsamen Eigenschaften in einem Basisfunktionsbaustein und leitet davon die verschiedenen Varianten ab, sodass nur noch die Unterschiede neu entwickelt werden müssen.“

Dirk Janssen betont, dass TwinCAT 3 auch die Inbetriebnahme weiter vereinfacht, da die Module fertig ausgetestet werden können. „Inbetriebnehmer arbeiten heute oft nur mit der Anpassungsschicht“, setzt er fort. „Vor Ort sind dann nur noch kleinere Veränderungen erforderlich, was sich einfach realisieren lässt. Die objektorientierten Erweiterungen bieten komfortable Möglichkeiten, um die vermehrte Modularisierung von Maschinen und deren Aggregaten auch im Steuerungsprogramm entsprechend abzubilden.“

Unterstützung von Multi-Core

„Die Multi-Core-Unterstützung durch TwinCAT 3 ermöglicht es, die Leistungsfähigkeit der Cores vollständig zu nutzen“, erläutert Dirk Janssen. „Dies hat signifikante Auswirkungen, da sich nun das Maschinenprogramm einfach auf verschiedene Cores aufteilen lässt.“ „Die Aufteilung wurde in unserem Haus bereits mit 16 Cores erfolgreich getestet“, schließt Ramon Barth an: „Dabei ermöglicht es die Hyperthreading-Funktion, virtuelle Cores zu bilden, die – im Test nur unwesentlich langsamer als echte Cores waren“. Auf einem Intel®-Xeon®-Serversystem mit zwei physikalischen Prozessoren und jeweils 6 Cores ließen sich problemlos 24 virtuelle Cores für Echtzeitcode nutzen.“

„Wir haben mit allen Prozessoren Benchmarks durchgeführt und die Ergebnisse haben gezeigt, dass die TwinCAT-Multi-Core-Unterstützung eine Performance-Steigerung linear zur Anzahl der CPU-Cores ermöglicht“, hebt Dirk Janssen hervor. „Dies beruht darauf, dass das Steuerungspro-



gramm sofort aus dem Speicher verfügbar ist und keine Wartezeiten erforderlich sind.“ „Bei einem Core fallen viele Unterbrechungen an, die Rechenleistung kosten“, führt Ramon Barth weiter aus: „Zuerst werden hochprioritäre Tasks bearbeitet. Bei Multi-Core werden Leistungsverluste, die durch die Synchronisierung der Cores entstehen, durch den Wegfall von Task-Kontextwechseln nahezu ausgeglichen.“ Die beiden Softwarespezialisten sind sich einig: „Je komplexer die Anwendung, desto größer ist der Nutzen.“

„Die Modularisierung und die Multi-Core-Technologie passen ideal zusammen“, betont Dirk Janssen. „Denn Multi-Core-Anwendungen ermöglichen es, dass die Module autark laufen und sich nicht gegenseitig beeinflussen.“ „Dies spart Rechenleistung“, ist Ramon Barth überzeugt. „Hinzu kommt, als weiterer Vorteil, dass die Programme bei der Modularisierung leichter getestet werden können.“

Komplexität reduzieren

„Die Beta-Phase von TwinCAT 3 wurde sehr intensiv genutzt, um die Software weiter zu verbessern“, stellt Dirk Janssen fest. „Dazu haben wir mit sehr unterschiedlichen Anwendern zusammen gearbeitet.“ Ramon Barth ergänzt hierzu: „Vertreter der klassischen Automatisierungstechnik haben viel Wert auf die objektorientierten Erweiterungen gelegt und ihre Modularisierungsvorstellungen entsprechend umgesetzt. Aber auch für die Matlab®/Simulink®-Unterstützung kam eine Reihe von Anregungen, zum Beispiel für den Austausch von Matlab®-Elementen.“ Aber auch die einfache Handhabung und Installation habe sich als Vorteil erwiesen, erklärt Ramon Barth: „Zum Beispiel war ein Tester aus dem Bereich komplexer Sensoren sehr zufrieden, der einen eigenen Hardwaretreiber geschrieben hatte.“ Bei den komplexen Sensorvorverarbeitungen kommt die Leistungsfähigkeit eines modernen PC-Systems mit Multi-Core-Unterstützung zum Tragen, da es spezielle und teure externe Messsysteme ersetzt. „Neu für uns waren Anwender, die komplett auf C/C++ als Automatisierungssprache setzen und keine SPS im Einsatz haben“, so Ramon Barth weiter. „Positives Feedback haben wir auch für die von uns aus der SPS-Welt übertragenen Debug-Möglichkeiten erhalten.“

Ramon Barth stellt heraus, dass TwinCAT 3 die Komplexität vereinfacht: „Es sind nur drei bis vier Maus-Clicks erforderlich, um die einzelnen Tasks auf verschiedene Cores zu verteilen.“ Anwender müssen nur ihren Code implementieren. Die Software sorgt für die komplette Infrastruktur und die Kommunikation über ADS von selbst. „Dies gilt ebenso für die

Integration von Matlab®/Simulink®“, ergänzt Dirk Janssen. „Diese läuft wie bei C++.“ „Um mit Matlab® ein TwinCAT-Systemmodul zu erstellen, wird kein spezielles Zielmodul benötigt – dies ist der PC“, so Ramon Barth. „Bisher kamen fertig definierte Regler zum Einsatz, die nicht mehr beeinflussbar waren“, betont Dirk Janssen. „Nun lassen sich individuelle Regler erzeugen und einbinden.“

Zur SPS/IPC/DRIVES 2011 wird die Version TwinCAT 3 releast. Die Auslieferung ist für Anfang 2012 vorgesehen.

www.beckhoff.de/TwinCAT3



Fragen aus der Praxis

Ramon Barth und Dr. Dirk Janssen, verantwortlich für die Software-Entwicklung bei Beckhoff, beantworten Fragen aus der Praxis.

ETZ: Inwieweit ist TwinCAT 3 auch für andere Steuerungshersteller offen?

Ramon Barth: Wir sehen durchaus das Interesse kleinerer Steuerungshersteller, ihren Code in TwinCAT ablaufen zu lassen. Damit können sie die vielfältigen Möglichkeiten nutzen, die unsere Softwareumgebung bietet: Darunter fallen auch die enge Integration in die PC-Welt und die Nutzung der EtherCAT-Kommunikation. Technisch ist das überhaupt kein Problem. Dies trifft nicht nur auf Steuerungshersteller zu. Auch Maschinenbauer haben ihr Know-how oft in die Programmiersprache C „gegossen“ und sehen nun die Möglichkeit, dieses in einer leistungsfähigen und offenen Umgebung zu nutzen.

ETZ: Wie kann der Quellcode geschützt werden?

Ramon Barth: Unser TwinCAT-Lizenzierungsmodell sieht vor, dass das Zielsystem identifiziert werden muss. Als Zusatzoption lässt sich der Anwender-Code verschlüsseln, sodass dieser nur auf einem Zielsystem bzw. nur auf einem geschützten System läuft. Übrigens spricht die zunehmende Bedeutung eines geschützten Quellcodes nicht gerade für Open-Source-Modelle.

ETZ: Was sind die Pläne für die strategische Weiterentwicklung von TwinCAT 3?

Ramon Barth: Die weitere Unterstützung zur Erstellung und Nutzung von bereits vordefinierten Softwaremodulen steht bei der Weiterentwicklung im Vordergrund. Wir haben in TwinCAT 3 ein einheitliches Beschreibungsformat für Softwaremodule eingeführt, das weiter ausgebaut und um entsprechende Tools ergänzt wird.

Dirk Janssen: Ende 2012 steht eine erweiterte Version von TwinCAT 3 an. Die Visualisierung, deren Erstellung wir in der Vergangenheit mit leistungsfähigen Datenschnittstellen unterstützt haben, wird hier eine wichtige Rolle spielen. Weiterhin bringen wir im nächsten Jahr ein Modul mit integrierter PC-Safety-PLC heraus. Darüber hinaus wird eine Authentifizierung bei der Kommunikation mit Steuerungen oder von Steuerungen untereinander integriert, das heißt, ein zusätzlicher Security-Level schützt noch besser vor Angriffen.

Ramon Barth: Das Laufzeitsystem wird 64-bit-fähig. Dies ist bisher nur die Entwicklungsumgebung, da im Runtime-Bereich kaum eine Nachfrage bestand.

ETZ: Inwieweit sind Sie mit dem Plan vorangekommen, Softwaremodule von Drittanbietern für TwinCAT mit anzubieten?

Dirk Janssen: Von der technischen Seite haben wir dieses Thema bereits angeschlossen. Die notwendigen Schnittstellen und Beschreibungsformate liegen vor und entsprechen exakt denen, die wir intern verwenden. Auf der Roadmap steht u. a. noch die Frage der Kommerzialisierung. Es bleibt unsere feste Zielsetzung, Softwaremodule von Drittanbietern in unser Produktportfolio aufzunehmen.

ETZ: Ist bei TwinCAT eine Eplan-Unterstützung vorgesehen?

Dirk Janssen: Bereits mit TwinCAT 2 ist es möglich, über Supplements, aus der Eplan-Welt, Variablen zu übernehmen. Selbstverständlich ist dies auch in der neuen Version 3 möglich. Wir sehen dies als Stand der Technik. Mithilfe eines Importwerkzeugs können E-CAD-Systeme angebunden werden.

ETZ: Können alte TwinCAT-Projekte mit der Version 3 gewartet werden?

Dirk Janssen: Es ist möglich, beide Versionen auf einem Rechner laufen zu lassen, sodass Servicetechniker die Tools auswählen können, die zur jeweiligen Runtime passen. Wir denken, dass Anwender ihre Zielsysteme mit der Version bearbeiten, in der die Programme auch erstellt wurden. Nur dies stellt sicher, dass nur das geändert wird, was der Anwender auch ändern will. Allerdings können bestehenden TwinCAT-Projekte in TwinCAT 3 importiert werden, um sie dann auf einer TwinCAT-3-Runtime unverändert und erweitert laufen zu lassen.

ETZ: Wie sehen die Lizenzmodelle für TwinCAT 3 aus?

Dirk Janssen: Es werden weiterhin die Laufzeitsysteme lizenziert und nicht die Programmierumgebungen. Für die Laufzeitsysteme haben wir ein gerechteres Lizenzmodell eingeführt, welches deutlich feingranularer ist. Basis-Runtime-Komponenten (Echtzeit, IO, PLC-Laufzeit,...) werden als Module betrachtet. Anwendermodule können sich in das Lizenzsystem integrieren und die Infrastruktur über Beckhoff mit nutzen. Die Lizenzierung erfolgt grundsätzlich pro Zielsystem.

Ramon Barth: Unsere Module sind übrigens auch nicht versionsgebunden.

ETZ: Ist kostenloses Testen noch vorgesehen?

Dirk Janssen: Selbstverständlich. Alle TwinCAT-Module unterstützen einen zeitlimitierten Testbetrieb. Es gibt keine funktionalen Einschränkungen.