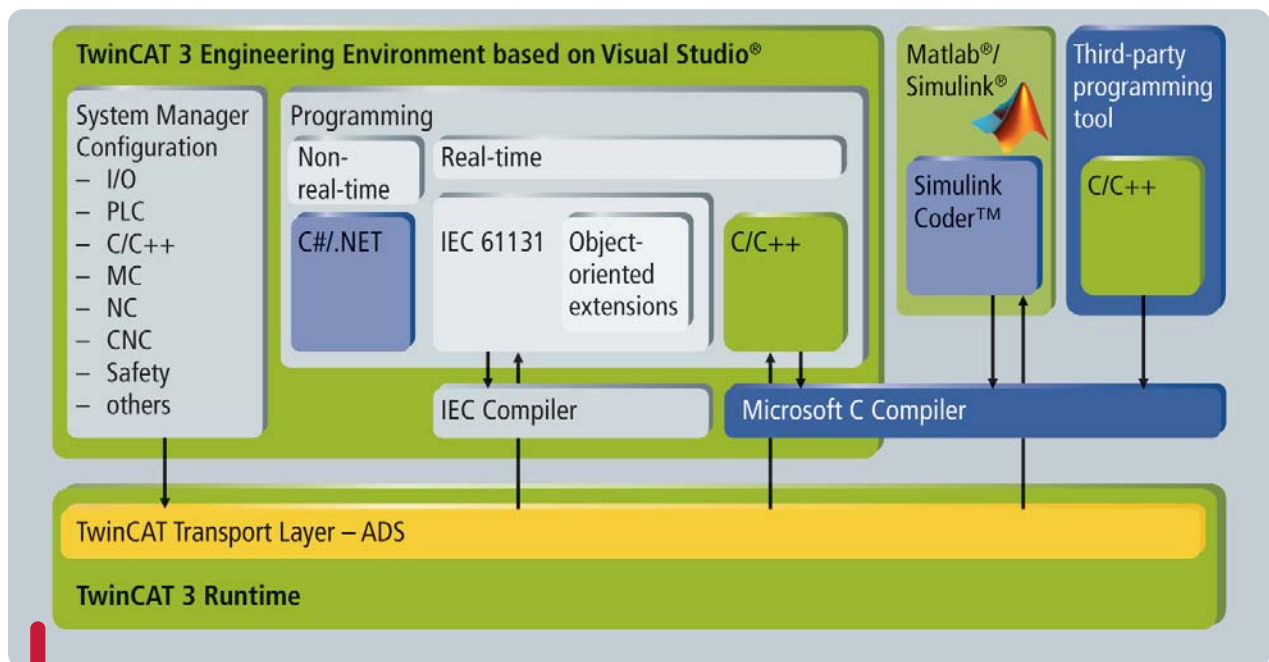


# Effizientes Engineering mit TwinCAT 3

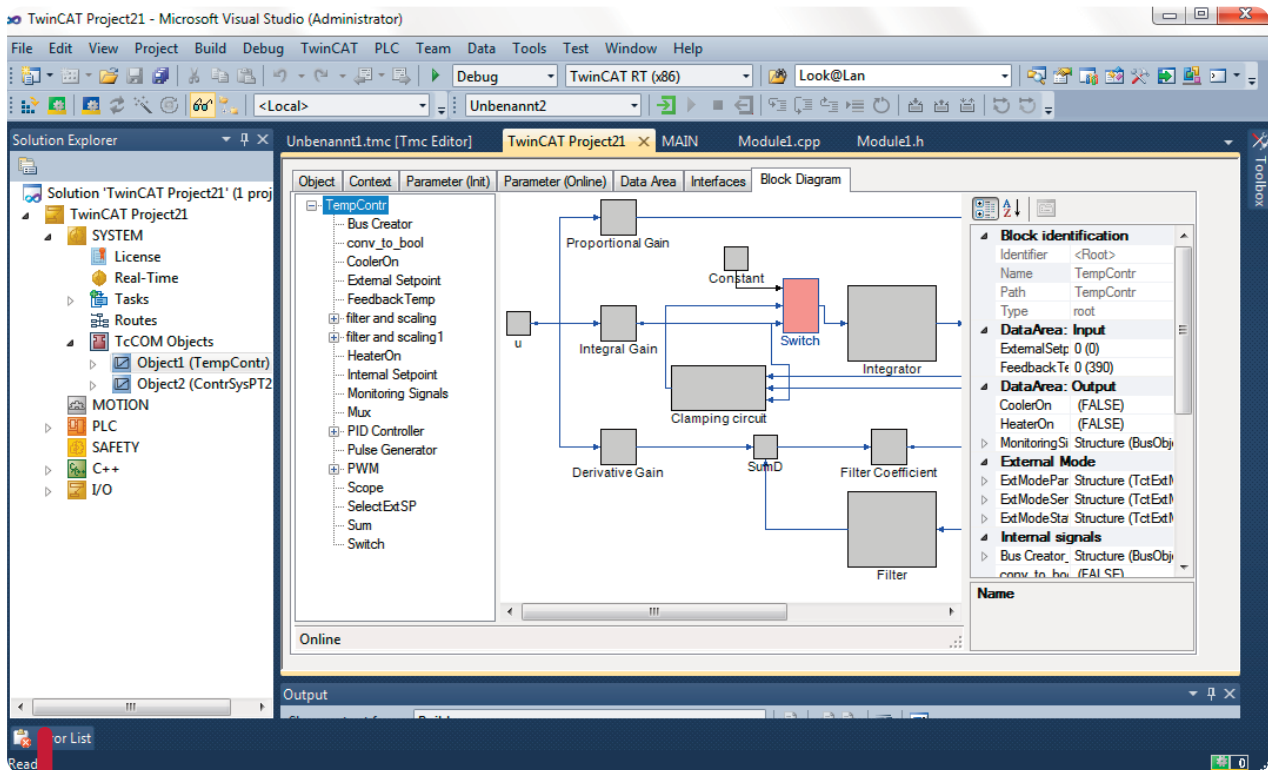


**Bild 1:** Die vollständige Integration von TwinCAT 3 in Visual Studio gibt dem Anwender die Möglichkeit, die Programmiersprachen IEC61131-3 und C/C++ parallel zu nutzen. Die erzeugten Objekte können, unabhängig von der Erstellungssprache, Daten austauschen und sich gegenseitig aufrufen. Der TwinCAT 3 System Manager wurde in die Entwicklungsumgebung integriert. Damit ist nur noch eine Software erforderlich, um Automatisierungsgeräte zu konfigurieren, parametrieren, programmieren und zu diagnostizieren.

Der Anteil der Steuerungssoftware an den Engineeringkosten bei der Automatisierung von Maschinen und Anlagen ist beträchtlich. Somit stellt die Reduzierung der Engineeringzeit für die Steuerungssoftware ein enormes Einsparpotenzial in Aussicht. Durch die von TwinCAT 3 unterstützte Parallelisierung von Engineeringprozessen sowie die Wiederverwendbarkeit von bereits existierenden Softwaremodulen lässt sich der Engineeringaufwand, bei gleichzeitiger Erhöhung der Qualität des Steuerungscodes, deutlich reduzieren. Die von

TwinCAT 3 durchgängig unterstützten, objektorientierten Techniken sowie die Bereitstellungen von zusätzlichen Sprachmitteln erlauben außerdem, das Prozesswissen von Maschinen- und Anlagenherstellern mithilfe moderner Softwaretechniken schnell und effizient umzusetzen. Durch die Einbettung in das Framework des Microsoft Visual Studio wird zudem eine Entwicklungsinfrastruktur zur Verfügung gestellt, wie sie zwar in der IT-Welt bereits lange als Stand der Technik angesehen wird, im Bereich der Automatisierung jedoch einmalig ist.

Die Anforderungen an Maschinen sind, in Hinblick auf ihre Produktivität, in den letzten Jahren enorm gewachsen. Als Folge dessen werden Maschinen immer komplexer. Bestand z.B. eine Maschine im Bereich der Holzbearbeitung, aus dem Jahr 1999, aus dreißig Aggregaten mit 400 I/O-Punkten und produzierte 25 bis 30 Teile in der Minute, besitzt eine Maschine neueren Typs bereits achtzig Aggregate, über 2.000 I/O-Punkte und produziert über 80 Teile je Minute (siehe Tabelle 1). Um dem gerecht zu werden, sind die Anforderungen nicht nur im Bereich der Steuerungshardware gestiegen, sondern auch in Bezug auf die Steuerungssoftware. Die PC-basierte Steuerungstechnik setzt hier in Bezug auf Rechnerperformance keine Grenzen; jedoch sind auch im Bereich der Steuerungssoftware neue Lösungen erforderlich. Zu den vorrangigen Anforde-



**Bild 2:** Markierung einer Fehlerstelle im Blockdiagramm

lungen, die bei neuen Softwarelösungen Berücksichtigung finden müssen, gehören:

- Die Steuerungssoftware muss performanter werden.
- Die Umsetzung komplexer Steuerungskonzepte muss mit den jeweils effizientesten Sprachmitteln möglich sein.
- Unterstützung der Objektorientierung, mit dem Ergebnis, wiederverwendbar, einfach zu wartender Softwaremodule.
- Die Softwaremodule müssen frei skalierbar sein: vom Abbilden eines Reglers bis hin zu einem kompletten Laufzeitsystem.
- Die durchgängige Verwendung von Interfaces in allen Beschreibungssprachen stellt die Interoperabilität der Softwaremodule sicher.
- Die Softwaremodule müssen ohne bzw. mit wenig Aufwand auf andere Plattformen portierbar sein.
- Die Softwarekonzepte müssen aktuelle IT-Technologien wie Multicore-CPU unterstützen.

Um diesen Ansprüchen sowie der Tatsache gerecht zu werden, dass für die Planung

einer Maschine oder Anlage immer weniger Zeit zur Verfügung steht, sind im Bereich des Engineerings einfache und durchgängige Lösungen erforderlich. Wichtige Faktoren, um dies zu erreichen, sind:

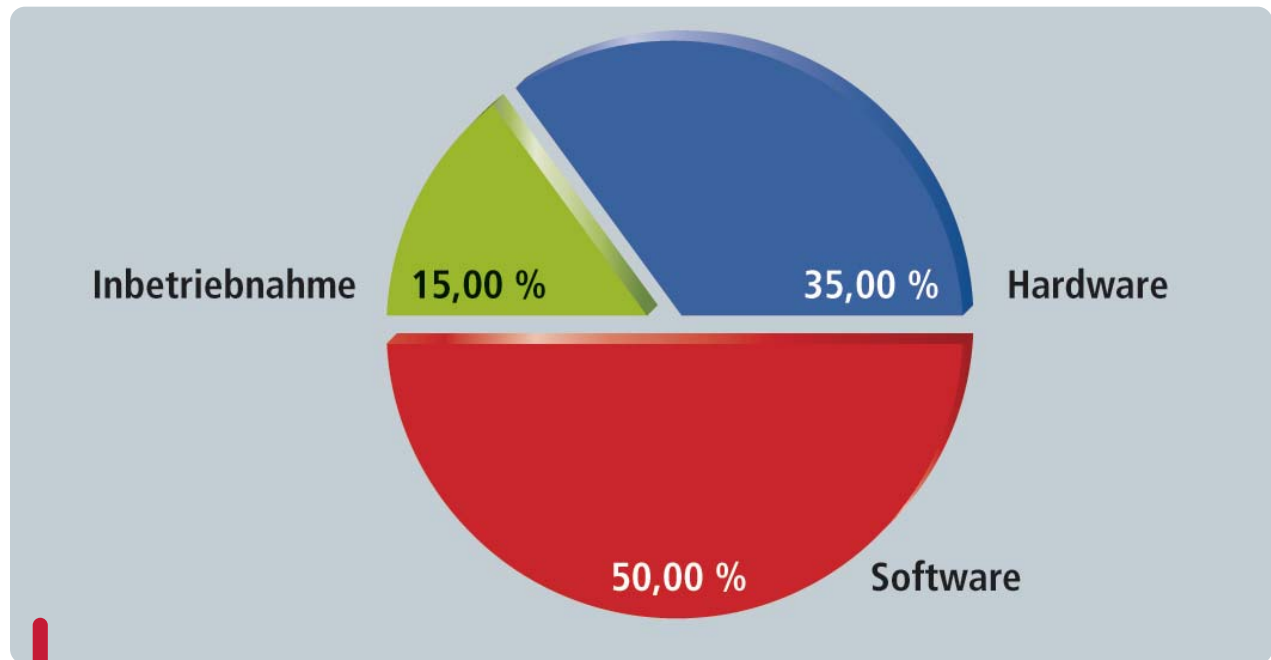
1. Unterstützung aktueller IT-Technologien (Multicore, 64-Bit- Betriebssysteme)
2. Unterstützung vielfältiger Sprachmittel (z.B. IEC61131-3, C++, Matlab/Simulink)
3. Bereitstellung eines einheitlichen Frameworks für alle zur Verfügung stehenden Sprachmittel
4. Durchgängige Unterstützung objektorientierter Techniken zur Erzeugung wiederverwendbarer, einfach zu erweiternder und zu wartender Module
5. Bereitstellungen von integrierten Simulationsmöglichkeiten, welche, ohne Mehraufwand (z.B. neues Kompilieren von Modulen bei Adressänderungen), eine Umschaltung in den Normalbetrieb der Anlage ermöglichen
6. Bereitstellung von durchgängigen und effizienten Debugging-Funktionen, welche es erlauben, in den erstellten Softwaremodulen, ohne zusätzliche Werkzeuge, effizient nach Fehlern zu suchen
7. Bereitstellung einfacher Möglichkeiten, die eine

automatische Generierung von Anlagenkonfigurationen bzw. von Steuerungscode erlauben

8. Bereitstellung der kompletten Entwicklungsinfrastruktur (z.B. Einbindung an Quellcodeverwaltungssysteme) in derselben Steuerungsumgebung

### Umsetzung mit TwinCAT 3

Um deutliche Performancesteigerungen der Steuerungssoftware zu erreichen, müssen, wie in (1) angeführt, aktuelle IT-Technologien unterstützt werden. TwinCAT 3 ermöglicht daher die aktive Ausnutzung der Ressourcen auf Multicore- und 64Bit-Architekturen. Das heißt, in TwinCAT 3 erstellte Softwaremodule können beispielsweise frei auf den zur Verfügung stehenden Kernen eines Multicore-Systems verteilt werden. Für jeden Kern kann, unabhängig von den anderen Kernen, eine eigene Basetime definiert werden. Basierend auf dieser, können



**Bild 3:** Wertanteile der Automatisierung 2005; Quelle/Literaturhinweis: HEINZE, R.: Mit Integration die Automatisierung automatisieren. openautomation, Vol. 9 (5), 2007, S. 18-21.

die dem System zur Verfügung stehenden 65.000 Tasks den einzelnen Kernen zugewiesen werden. Die in Punkt (2) geforderte Unterstützung verschiedener Sprachmittel soll ermöglichen, dass Aufgabenstellungen mit der jeweils für dieses Problem effizientesten Sprache umgesetzt werden können. Aus diesem Grund stehen in Twincat 3, neben den Sprachen der IEC61131-3, auch die Sprachen C/C++ sowie Matlab/Simulink für die Erstellung von Softwaremodulen zur Verfügung. In der Praxis heißt dies, dass die Beschreibung eines Ablaufes mithilfe der IEC61131-3, die Beschreibung eines komplexen Algorithmus, unter Verwendung von C++, sowie die Ausführung eines, auf Basis von Matlab/Simulink erzeugten, Reglers in ein und demselben Softwareprojekt erfolgen können. Die Ausführung dieser Module erfolgt in Twincat 3 in derselben Laufzeit. Durch die in (3) geforderte Verwendung eines durchgängigen Frameworks ist es darüber hinaus möglich, dass die Module optimal miteinander interagieren können, d.h. Daten austauschen, sich gegenseitig aufrufen, Interfaces der anderen Module ansprechen etc. Wie in (4) angeführt, kommen in Twincat 3 bei der Erstellung dieser Module durchgängig objektorientierte Techniken zum Einsatz. Ziel ist es, Objekte so zu kapseln, dass einfach

zu wartende, erweiterbare Module entstehen, die wiederverwendbar sind. Dies ermöglicht, unter anderem, die mehrfache Instanziierung und Verknüpfung eines aus Matlab/Simulink generierten Reglers (ohne diesen mehrfach zu kompilieren) im selben Softwareprojekt. Auch in den Sprachen der IEC61131-3 stehen neue objektorientierte Sprachmittel, wie Interfaces, Vererbung oder Methoden zur Verfügung. Darüber hinaus ist nunmehr auch das dynamische Erzeugen von Objekten zur Laufzeit in den Sprachen der IEC61131-3 erlaubt. Für eine effiziente Entwicklung bzw. für den Test und die Inbetriebnahme von Softwaremodulen ist, wie in (5) angeführt, die Einbettung verschiedener Simulationsmöglichkeiten erforderlich. Auch dies wird in Twincat 3 durch das modulare Konzept ermöglicht. Neben den Softwaremodulen, die die eigentliche Steuerungslogik enthalten, können weitere Module in das Projekt eingebunden werden, um ganze Anlagenteile zu simulieren. Diese Module können nach Bedarf zu- oder abgeschaltet werden, sodass auch eine Teilsimulation der Maschine oder Anlage möglich ist. Darüber hinaus bietet die Anbindung an die Werkzeuglandschaft von Matlab/Simulink auch die Möglichkeit, Entwicklungsmethoden, wie Rapid Control Prototyping, zu unterstützen. Die Entwick-

lung der einzelnen Softwaremodule erfolgt hierbei in mehreren Schritten:

- Generierung des Modells der zu steuernden Hardware
- Generierung des Softwaremoduls
- Test des Moduls im Simulationsbetrieb (HIL – Hardware in the Loop)
- Inbetriebnahme des Moduls
- Inbetriebnahme des Gesamtsystems

Parallel zur Modellierung und Erzeugung des Simulationsmoduls für die Hardware kann auch ein Simulationsmodul der Steuerungssoftware erzeugt werden, welches bereits den Ablauf, noch nicht aber die fertigen Algorithmen enthält. Ziel dieses Vorgehens ist der Test der Anlagenhardware, ohne dass die Steuerung bereits vollständig zur Verfügung steht (SIL – Software in the Loop). Zum Auffinden von Fehlern (6) während der Entwicklung von Modulen müssen in allen unterstützten Beschreibungssprachen effiziente Debugging-Funktionalitäten zur Verfügung stehen. Gehört die Anzeige von Onlinewerten bei Editoren der IEC61131-3-Sprachen bereits zum Standardfunktionsumfang, ist dies bei Editoren für die Sprache C++ bisher nicht möglich. Auch hierbei ist Twincat 3 das erste Werkzeug, das dies direkt in der Entwicklungsumgebung ermöglicht. Eine weitere Standardfunktionalität im Bereich Debugging

## Eckdaten von Holzbearbeitungsmaschinen desselben Herstellers

	1999	2011
CPU	1 x Intel® Pentium® 4	1 x Intel® Core™ 2 Duo
Anzahl Aggregate	30	> 80
Anzahl I/O-Punkte	400	2.000
Anzahl Servoachsen	20	> 100
Geschwindigkeit	30 m/min	> 80 m/min

sind Breakpoints. Wenngleich die Verwendung von Breakpoints in den Sprachen der IEC61131-3 oder C++ bereits Stand der Technik ist, gilt dies nicht für das 'Blockschaltbild' eines aus Matlab/Simulink generierten Moduls. Ähnliches gilt für das Lokalisieren von Fehlerstellen. Werden diese, für aus Matlab/Simulink generierte Module, typischerweise in Form von Zeilennummern im Code wiedergegeben, ermöglicht TwinCAT 3 hier die grafische Abbildung der Fehlerstelle im Blockschaltbild. Mit anderen Worten, ermöglicht TwinCAT 3 durchgängig die Verwendung von Debugging-Funktionalitäten, welche in allen zur Verfügung gestellten Sprachmitteln identische Möglichkeiten, wie Online-Monitoring, Breakpoint, Callstack, Disassembly-Window etc., bereitstellen. Eine weitere Möglichkeit, um die Effizienz des Engineerings deutlich zu steigern, ist die in (7) angeführte Verwendung von Automatismen, welche die Konfiguration der Steuerungshardware oder Teile des Steuerungsprogrammes automatisch generieren. Auch hierfür bietet TwinCAT 3 die optimale Voraussetzung: Erreicht wird dies zum einen durch die Unterstützung von Standardformaten, z.B. die Verwendung des PLCopen-XML-Formates als Datenaustauschformat von SPS-Projekten, und durch die Abbildung der kompletten

Steuerungskonfiguration mittels XML-basierten Datenformaten. Zum anderen steht ein Beckhoff Automation Interface (API) bereit, mit welchem auf die komplette Funktionalität von TwinCAT 3 zugegriffen werden kann. Wie in (8) angeführt, ist auch die Bereitstellung einer durchgängigen Entwicklungsumgebung ein wesentlicher Faktor, um das Engineering von Maschinen und Anlagen effizienter zu gestalten. Hierzu gehören u. a. die Anbindung der Entwicklungsumgebung an Quellcodeverwaltungs-, Bug-Tracking- und Projektmanagement-Werkzeuge. Durch die Einbettung von TwinCAT 3 in den Rahmen des Visual Studio 2010 sowie die Unterstützung von dessen nativen Schnittstellen ist diese Entwicklungsumgebung optimal gegeben.

### Nutzen für den Maschinenbauer

Um die Komplexität moderner Maschinen zu beherrschen und gleichzeitig der Forderung nach immer größerer Flexibilität und immer kürzeren Planungszeiten nachzukommen, geht der Trend bei Maschinen und Anlagen hin zu modularen, wiederverwendbaren Softwarestrukturen. TwinCAT 3 ermöglicht, die modularen Strukturen der Maschine, unabhängig von der gewählten Beschreibungssprache, direkt in der Soft-

ware abzubilden. Dies beinhaltet sowohl die Abbildung der Maschinenmodule, in Form von gekapselten Softwareobjekten (Klassen), als auch der Objektschnittstellen, mithilfe von Interfaces. Ein wesentlicher Vorteil dieses durchgängigen objektorientierten Ansatzes ist die Steigerung der Wiederverwendbarkeit. Einzelne Softwaremodule lassen sich sowohl innerhalb desselben Softwareprojektes – z.B. bei der mehrfachen Instanziierung eines aus Matlab/Simulink generierten Reglers – als auch für neue Projekte einfach wiederverwenden. Objektorientierte Programmierung führt zudem zu einer verbesserten Lesbarkeit des Steuerungscode, womit eine einfache Erweiterbarkeit und die Qualitätssteigerung des Steuerungscode einhergehen. Auf Basis dieser Kapselung ist – eine Definition entsprechender Schnittstellen vorausgesetzt – eine Parallelisierung des Software-Engineerings möglich. Zusätzlich unterstützt wird dies durch die Anbindung an moderne Entwicklungsinfrastrukturen, wie Quellcodeverwaltung, Bugtracking etc. Auch die in (8) angeführte Unterstützung von Automatismen, im Hinblick auf die Generierung von Steuerungskonfigurationen und Programmteilen, führt zu einer deutlichen Effizienzsteigerung. ■

[www.beckhoff.com](http://www.beckhoff.com)



Autor: Knut Güttel, Produktmanager TwinCAT, Beckhoff Automation



Autor: Dr. Josef Papanfort, Produktmanager TwinCAT, Beckhoff Automation