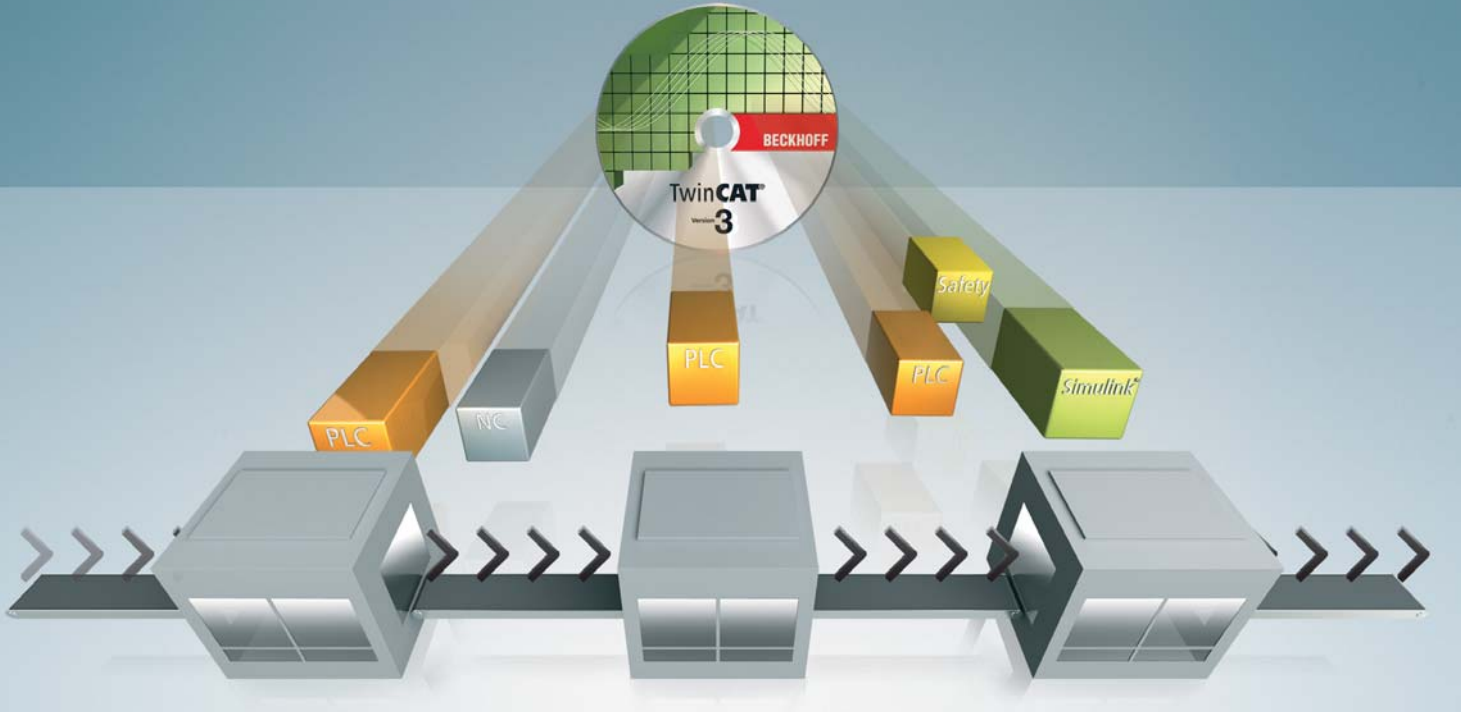


Modulare Automatisierung mit TwinCAT 3



Moderne Maschinen sind aus einzelnen, hierarchisch angeordneten Funktionsmodulen oder Aggregaten aufgebaut. Diese Aggregate können weitgehend unabhängig voneinander entwickelt und getestet werden. Über standardisierte Schnittstellen können sie dann an der konkreten Maschine mit anderen interagieren. Eine ähnliche Modularisierung ist auch in der Steuerungssoftware notwendig, sodass jedes Aggregat über eine eigene „Steuerung“ verfügt, die wiederum über standardisierte Schnittstellen mit den anderen Steuerungen kommuniziert.

TwinCAT 3 bietet durch den modularen Aufbau der erweiterten Runtime und die für viele Anwendungsfälle bereits vordefinierten Schnittstellen die Möglichkeit, die unterschiedlichen Steuerungen der Maschinenaggregate gemeinsam auf der zentralen Steuerungshardware der Maschine zu instanzieren. Diese können unabhängig voneinander und in verschiedenen Programmiersprachen erstellt werden. Eine große Auswahl bereits vorhandener oder selbst entwickelter Basismodule bildet dabei einen Baukasten, aus dem sich leicht neue Anwendungen erstellen lassen.

Historie

Der Ansatz, die klassischen Automatisierungsgeräte, wie speicherprogrammierbare Steuerungen (SPS) und Motion-Steuerungen, in Software auf einer leistungsfähigen Standard-Hardware zu implementieren, ist seit vielen Jahren Stand der Technik und wird inzwischen von vielen Anbietern verfolgt. Die Vorteile sind vielfältig; der wichtigste Vorteil ist aber sicherlich, dass die Software von der Hardware weitestgehend unabhängig ist und dadurch die Leistungsfähigkeit der Hardware an die Anwendung angepasst werden kann. Zusätzlich profitiert die Anwendung automatisch von der allgemeinen Weiterentwicklung der Hardware. Dies trifft insbesondere auf PC-Hardware zu, die nach wie vor dramatische Leistungszuwächse mit sich bringt. Für den Anwender ist auch noch die relative Unabhängigkeit von einem Anbieter wichtig, die diese Trennung von Soft- und Hardware ebenfalls zur Folge hat.

Da die SPS und die Motion-Steuerung – und eventuell weitere Automatisierungskomponenten – bei diesem Ansatz weiterhin als eigenständige, logische Einheiten existieren, ändert sich an der Applikationsarchitektur verglichen zur klassischen Automatisierungstechnik recht wenig. Die SPS bestimmt den logischen Ablauf der Maschine und beauftragt die Motion-Steuerung, bestimmte Achsfunktionen auszuführen. Durch die gestiegene Leistungsfähigkeit der Steuerungen und die Möglichkeit, höherwertige Programmiersprachen zu verwenden (IEC 61131-3), können so auch komplexe Maschinen automatisiert werden.

Modularisierung

Um die Komplexität moderner Maschinen zu beherrschen und gleichzeitig die notwendigen Engineering-Aufwendungen zu reduzieren, haben viele Maschinenbauer begonnen, ihre Maschinen zu modularisieren. Einzelne Funktionalitäten, Aggregate oder Maschineneinheiten werden dabei als möglichst eigenständige Module betrachtet, die dann über einheitliche Schnittstellen in das Gesamtsystem eingebettet werden. Im Idealfall ist eine Maschine dann hierarchisch aufgebaut, wobei die untersten Module einfachste, immer wiederverwendbare Basiselemente darstellen. Zusammengefügt bilden sie immer komplexere Maschineneinheiten, bis auf der obersten Ebene die gesamte Maschine entsteht.

Bei der steuerungstechnischen Umsetzung der Maschinen-Modularisierung werden unterschiedliche Ansätze verfolgt. Grob unterteilen lassen sie sich in einen dezentralen und in einen eher zentralen Ansatz. Beim dezentralen Ansatz erhält jedes Maschinenmodul eine eigene Steuerung, die die SPS- und eventuell auch Motion-Funktionalität des Moduls bestimmt. Die einzelnen Module können autark voneinander in Betrieb genommen, gewartet und relativ unabhängig skaliert werden. Die notwendigen Interaktionen zwischen den Steuerungen werden über Kommunikationsnetzwerke (Feldbusse oder Ethernet) koordiniert und über entsprechende Profile standardisiert.

Der zentrale Ansatz zieht die Steuerungsfunktionen aller Module in der gemeinsamen Steuerung zusammen und nutzt nur sehr geringe vorverarbeitende Intelligenz in den dezentralen I/O-Geräten. Die Interaktionen können viel direkter innerhalb der zentralen Steuerung erfolgen; dadurch werden die Kommunikationswege deutlich kürzer, Totzeiten entfallen und die gemeinsame Nutzung der Steuerungshardware senkt die Gesamtkosten.

Der zentrale Weg hat aber auch den Nachteil, dass die notwendige Modularisierung der Steuerungssoftware nicht automatisch vorgegeben wird. Die Möglichkeit, in der zentralen Steuerung auf jede beliebige Information aus anderen Teilen des Programms zugreifen zu können, behindert gleichzeitig die Modulbildung und die Wiederverwendbarkeit dieser Steuerungssoftware in anderen Applikationen. Da kein Kommunikationskanal zwischen den Steuerungseinheiten vorhanden ist, bleibt auch eine entsprechende Profilbildung und Standardisierung der Steuerungseinheiten häufig auf der Strecke.

Von beiden das Beste

Die ideale Steuerung für modulare Maschinen nimmt Anleihen sowohl von der dezentralen als auch von der zentralen Steuerungsarchitektur. Als Steuerungshardware kommt „natürlich“ eine zentrale, leistungsfähige, möglichst allgemeine Rechner-Plattform zum Tragen. Die Vorteile der zentralen Steuerungstechnik: geringere Gesamtkosten und die Möglichkeit, ohne Kommunikationsverluste auf alle Informationen des Gesamtsystems zugreifen zu können, sind entscheidende Argumente.

Die bereits angerissenen Vorteile des dezentralen Ansatzes lassen sich durch entsprechende Modularisierung der Steuerungssoftware auch in der Zentrale verwirklichen. Statt ein großes, komplexes SPS-Programm und eine NC mit vielen Achsen laufen zu lassen, können viele kleine „Steuerungen“ in einer gemeinsamen Runtime auf derselben Hardware relativ unabhängig voneinander koexistieren. Die einzelnen Steuerungsmodule werden gekapselt und bieten über standardisierte Schnittstellen ihre Funktionen nach außen an oder nutzen entsprechende Funktionen anderer Module oder der Runtime.

Eine sinnvolle Profilbildung erfolgt durch Festlegung dieser Schnittstellen und Standardisierung der entsprechenden Parameter und Prozessdaten. Da die einzelnen Module innerhalb einer Runtime ausgeführt werden, sind auch direkte Aufrufe in andere Module möglich – wiederum über entsprechende standardisierte Schnittstellen. Die Modularisierung kann daher frei an sinnvollen Grenzen erfolgen, ohne Rücksicht auf Kommunikationsverluste nehmen zu müssen.

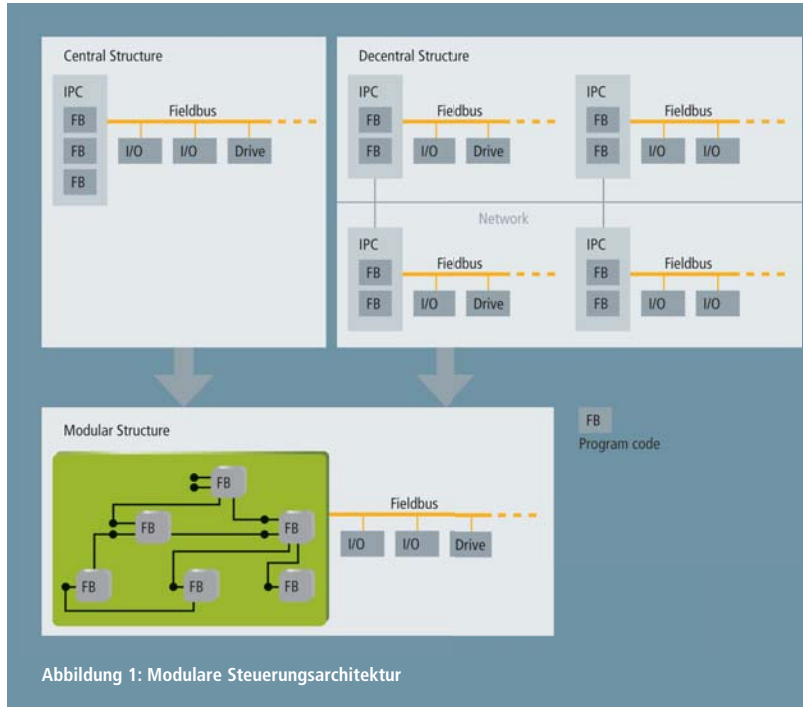


Abbildung 1: Modulare Steuerungsarchitektur

Während der Entwicklung oder Inbetriebnahme einzelner Maschinenmodule können die dazugehörigen Steuerungsmodule auf einer beliebigen Steuerungshardware mit entsprechender Runtime erstellt und getestet werden. Fehlende Anbindungen zu anderen Modulen können in dieser Phase emuliert werden. An der kompletten Maschine werden sie dann gemeinsam auf der zentralen Runtime instanziiert, diese muss nur so dimensioniert sein, dass die Anforderungen aller instanziierten Module (Speicher, Tasks und Rechenleistung) erfüllt werden.

TwinCAT-3-Module

Ein TwinCAT-Modul besteht aus einer Reihe von formal definierten Eigenschaften, die teilweise vorgeschrieben und teilweise optional sind. Die Eigenschaften sind insoweit formalisiert, dass eine allgemeine Nutzung sowohl untereinander als auch von außen möglich wird. Für die Konfiguration der Module und deren Beziehungen untereinander, bringt jedes Modul eine Modulbeschreibungdatei (XML-Format) mit. Wird ein Modul in der TwinCAT-Runtime instanziiert, dann meldet es sich bei einer zentralen System-Instanz, dem Modul-Manager, an. Dadurch wird es für andere Module und auch für allgemeine Tools erreichbar und parametrierbar. Module können unabhängig voneinander kompiliert und daher auch unabhängig voneinander entwickelt, getestet und gepflegt werden. Sie können sehr einfach aufgebaut sein, enthalten z. B. nur eine kleine Funktion, wie einen Tiefpassfilter; sie können aber auch intern sehr komplex sein und z. B. die komplette Steuerung eines Maschinenaggregates beinhalten. Der Einsatzfall von Modulen ist vielfältig; sämtliche Aufgaben eines Automatisierungssystems können werden in Module verpackt. So wird nicht unterschieden, ob die Module primär Basisfunktionalitäten eines Automatisierungssystems, wie Echtzeit-Tasks, Feldbus-Treiber oder ein SPS-Laufzeitsystem, darstellen oder eher anwender- und applikationsspezifische Algorithmen zur Steuerung oder Regelung eines Maschinenaggregates.

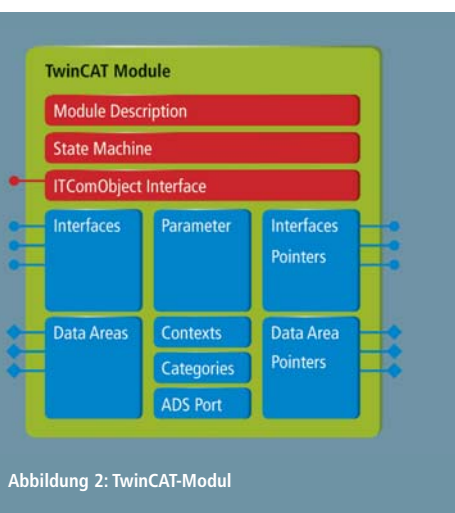


Abbildung 2: TwinCAT-Modul

Eigenschaften der Module

In Abbildung 2 ist ein allgemeines TwinCAT-Modul mit seinen wesentlichen Eigenschaften dargestellt. Die roten Blöcke definieren die vorgeschriebenen und die blauen die optionalen Eigenschaften.

Neben einer Modulbeschreibung, die für die Konfiguration der Module untereinander wichtig ist, besitzt jedes Modul eine allgemeine State-Machine, über die die Initialisierung, Parametrierung und Verlinkung der Module gesteuert wird. Das vorgeschriebene Interface `ITComObject` regelt den Zugriff auf die State-Machine und die Parameter der Module. Über weitere Interfaces wird die eigentliche Funktion des Moduls genutzt. Es existiert bereits eine große Auswahl vordefinierter Interfaces, die allgemeine Funktionen beschreiben – z. B. den zyklischen Aufruf der internen Modullogik. Weitere standardisierte, aber auch applikationsspezifische Interfaces können hinzugefügt werden. Über Interface-Pointer ist der Zugriff auf Interfaces anderer Module möglich.

Die Data-Areas beschreiben Datenbereiche, die von einem Modul für andere zur Verfügung gestellt werden. Hierdurch können Module auf standardisiertem Wege effizient auf gemeinsame Daten zugreifen. Der Zugriff auf die Daten eines anderen Moduls erfolgt dabei über die Data-Area-Pointer.

System-Module

Die TwinCAT-Runtime stellt auch eine Reihe von sogenannten System-Modulen zur Verfügung, die Basisdienste der Runtime für andere Module bereitstellen. Diese System-Module besitzen eine feste, konstante Object-ID und können dadurch von anderen Modulen erreicht werden. Ein Beispiel für ein System-Modul stellt das Echtzeit-System dar, das über sein `ITCRealTime`-Interface die Basisdienste des Echtzeit-Systems zur Verfügung stellt – z. B. das Erzeugen von Echtzeit-Tasks. Der ADS-Router ist ebenfalls als System-Modul implementiert, sodass andere Module ihren ADS-Port dort anmelden können.

Erstellen von Modulen

Module können sowohl in C++ als auch in IEC 61131-3 erstellt werden. Dazu werden die objektorientierten Erweiterungen der in TwinCAT 3 integrierten SPS genutzt. Module aus beiden Welten können über Interfaces genauso interagieren wie reine C++-Module untereinander. Mit Hilfe der objektorientierten Erweiterungen in der SPS werden die gleichen Interfaces genutzt wie in C++. Auch die SPS-Module melden sich beim Modul-Manager an und sind darüber entsprechend erreichbar. Die Komplexität der Module ist auch bei den SPS-Modulen variabel; ob nur ein kleines Filter-Modul erzeugt wird oder ein komplettes SPS-Programm in ein Modul verpackt wird, ist unerheblich.

In TwinCAT 3 ist sogar jedes SPS-Programm per Automatismus ein Modul im Sinne der TwinCAT-Module. Jedes klassische SPS-Programm wird automatisch in ein Modul verpackt und meldet sich beim Modul-Manager und bei einem oder mehreren Task-Modulen an. Der Zugriff auf die Prozessdaten eines SPS-Moduls (z. B. das Mappen zu einem Feldbus-Treiber) wird ebenfalls über die definierten Data-Areas geregelt. Für den SPS-Programmierer bleibt dieses Verhalten solange unsichtbar, bis er sich entschließt, explizit Teile des SPS-Programms als TwinCAT-Module zu definieren, um sie entsprechend flexibel einsetzen zu können.

TwinCAT-3-Runtime

Die TwinCAT-Runtime bietet eine Software-Umgebung, in der TwinCAT-Module geladen, ausgeführt und verwaltet werden. Sie bietet zusätzlich Basisfunktionen, um Ressourcen des Systems nutzen zu können (Speicher, Tasks, Feldbus- und Hardware-Zugriffe etc.). Die einzelnen Module müssen nicht in einem Kompilierzusammenhang stehen und können daher unabhängig voneinander sein und von unterschiedlichen Herstellern stammen.

Eine Reihe von System-Modulen werden beim Start der Runtime automatisch geladen, sodass deren Eigenschaften anderen Modulen zur Verfügung stehen. Der Zugriff auf Eigenschaften der System-Module findet aber auf die gleiche Weise statt, wie auf Eigenschaften normaler Module, sodass es für die Module unwichtig ist, ob die jeweilige Eigenschaft von einem System-Modul oder einem normalen Modul bereitgestellt wird.

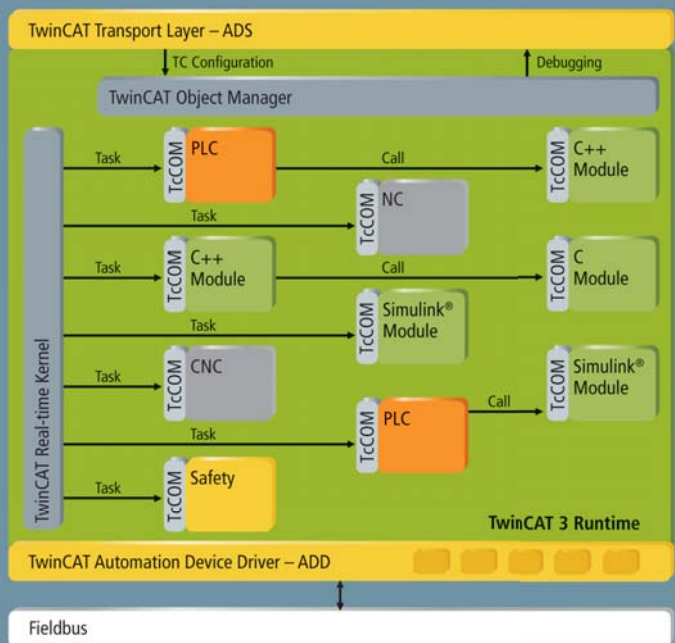


Abbildung 3: TwinCAT-3-Runtime

Erweiterbarkeit

TwinCAT 3 ist intern selbst in Form von Modulen entwickelt worden. Eigenschaften, wie Feldbustreiber oder Motion-Funktionen, sind nach den gleichen Regeln und Konventionen erstellt worden, wie sie auch für eher anwendungsorientierte Module gelten. Neue Funktionen, Algorithmen und auch die Unterstützung weiterer Kommunikationsprotokolle können so leicht dazu gebunden werden und ergänzen oder ersetzen die im Standardumfang vorhandenen Module.

Die Unterstützung eigener Hardware (Feldbuskarte, Frame-Grabber etc.) kann mit entsprechenden Treiber-Modulen realisiert werden und steht dann gleichberechtigt mit den in TwinCAT 3 vorhandenen Treibern zur Verfügung.

www.beckhoff.de/TwinCAT3



Dr. Dirk Janssen, Leiter
Software-Entwicklung System, CNC
und I/O bei Beckhoff Automation

Highlights

Komfortable Entwicklung

Die Entwicklung und der Test neuer Steuerungsmodule können unabhängig von der spezifischen Gesamtlösung erfolgen. Der Entwickler kann dabei auf eine Vielzahl vorhandener Module zurückgreifen und sie in einer modernen, auf Microsoft Visual Studio® basierenden Entwicklungsumgebung zu neuen Modulen oder ganzen Steuerungsprogrammen zusammensetzen. Sowohl die Automatisierungssprachen der IEC 61131-3 als auch C und C++ stehen zur Verfügung und können auch gleichberechtigt nebeneinander genutzt werden. Zusätzliche Entwicklungswerkzeuge, wie z. B. Matlab®/Simulink®, können ebenfalls eingesetzt werden und TwinCAT-3-Module erzeugen.

Leistungsfähige Steuerung

Durch die Interaktion der TwinCAT-3-Steuerungsmodule gemeinsam auf einer Steuerungsplattform und definierten Schnittstellen untereinander, kann die volle Performance eines Industrie- oder Embedded-PCs ohne Kommunikations- oder Synchronisierungsverluste genutzt werden. TwinCAT 3 kann alle Cores moderner CPUs nutzen und ist auch auf 64-Bit-Systemen einsatzfähig.

Wirtschaftliche Gesamtlösung

Der Einsatz vorentwickelter und getesteter Steuerungsmodule reduziert signifikant den Engineering-Aufwand für neue Maschinen und Anlagen. Die Steuerungsapplikationen lassen sich zu einem großen Teil aus vorhandenen Modulen zusammensetzen, sie werden allein durch Parametrierung und Verschaltung untereinander an die aktuelle Aufgabe angepasst. Die Steuerungsmodule werden auf einer zentralen, allgemeinen Steuerung instanziiert und ermöglichen dadurch den Einsatz kostengünstiger Hardware. Auf unterlagerte, dezentrale Steuerungshardware kann verzichtet werden.