

# Echt echtzeitfähig

Ramon Barth (Abb.), Leiter der Software-Entwicklung bei Beckhoff Automation, erklärt, wie Beckhoff Microsoft-Betriebssysteme echtzeitfähig macht und verrät im Interview schon neueste Features der kommenden TwinCAT 3.1-Version.



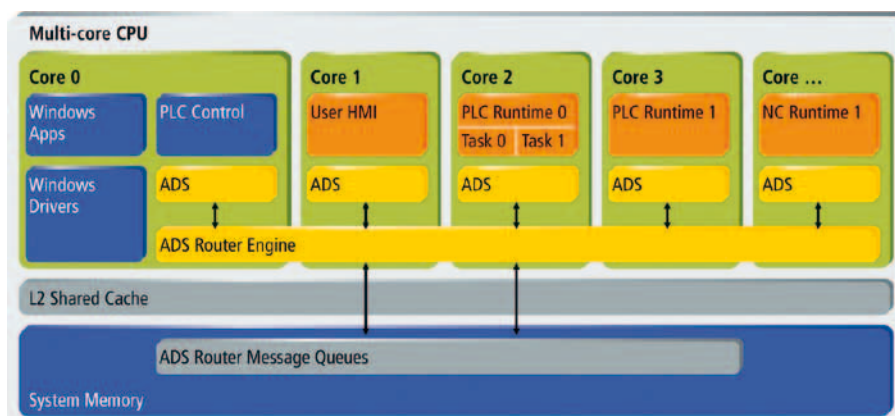
**Herr Barth, weshalb nutzt Beckhoff die Betriebssysteme von Microsoft als TwinCAT-Plattform?** Ramon Barth: Bei Entwicklungen in der Steuerungstechnik hat Beckhoff schon immer allgemein verfügbare Mainstream-Produkte aus dem IT-Bereich genutzt und für die Automatisierungstechnik angepasst. Zum einen bleiben so die Projekte bezahlbar und zum anderen kennen sich die meisten Anwender mit diesen Systemen aus. Es gibt wahrscheinlich keinen Compiler, der intensiver getestet wurde als der Microsoft C++

Compiler. Vielleicht heutzutage noch der GNU++ Compiler, aber zu Beginn unserer Entwicklung vor mehr als 25 Jahren waren die Microsoft-Tools am weitesten verbreitet. Aufgrund der Entwicklungen, die wir damals – wie heute – am IT-Markt sehen, haben wir uns entschieden, auf Mainstream-Produkte zu setzen und dort unsere Erweiterungen und unser Know-how einzubringen, um dann in Summe sehr performante und preisgünstige Gesamtsysteme liefern zu können.

**Wie kam es zur Kooperation zwischen Beckhoff und Microsoft?** 1996 hatten wir ein größeres Projekt, in das auch Microsoft involviert war. Microsoft hatte vorgeschlagen, die Umsetzung mit Windows NT zu machen und angeboten, uns dabei beratend zu unterstützen. Es gab dann ein historisches Treffen der Entwickler in einem Microsoft-Labor in Paris zur Anpassung unserer Software an Windows NT. Windows NT konnte damals schon echtzeitfähig auf Interrupts antworten. Das liegt an der internen Struktur von NT, die bis heute erhalten ist. Also mussten wir nichts an Windows NT selbst verändern, um eine Echtzeitfähigkeit zu erreichen, sondern lediglich unsere Echtzeiterweiterung entwickeln. So kam es zur Kooperation mit Microsoft.

**Hätte es Alternativen zu Windows gegeben?** Damals gab es keine Alternativen zu Microsoft; heute ist die Situation eine andere, z.B. durch Linux. Aber bevor wir noch einmal eine Neuentwicklung starten würden, müsste uns Linux schon einen echten Vorteil bieten. Den sehen wir aber derzeit nicht. Erschwerend kommt bei OpenSource-Systemen noch die Patentproblematik hinzu.

**Läuft TwinCAT schon auf Windows 8?** Ja, so ist es. TwinCAT läuft auf allen Windows-Versionen, die von Windows NT abgeleitet wurden, inklusive Windows 8. Für die aktuelle Windows 8-Version, haben wir allerdings einige Anpassungen an TwinCAT vorgenommen, um die veränderte Hardware-Welt zu unterstützen.



◀ Die TwinCAT-3-Laufzeitumgebung unterstützt die Multi-Core-Technologie moderner CPUs, sodass funktionale Einheiten, wie HMI, PLC-Runtime oder Motion Control auf dedizierte Kerne verteilt werden können.

**Wie haben Sie Windows NT echtzeitfähig gemacht und was ist das Besondere daran?**

Windows NT kann echtzeitfähig Interrupts verarbeiten und PC-Control ist zeitgesteuert, das heißt, wir benutzen einen Timer-Interrupt. Dieser steuert unseren Echtzeit-Scheduler, unser TwinCAT. Die Tasks werden vom Scheduler in Echtzeit verwaltet, also zur richtigen Zeit mit der passenden Priorität umgeschaltet. Das Ganze wird getrieben von einem zentralen Timer-Interrupt, der in Echtzeit kommt. Der Scheduler ist so aufgebaut, dass bis zu 64 Tasks geschaltet werden können. Darüber hinaus können wir die Basiszeit vom Timer verändern, um Ressourcen zu sparen. Das heißt, wenn eine Task über 100 µs Zykluszeit benötigt, dann stellen wir einfach das Gesamtsystem auf 100 µs um. Wenn die schnellste Task eine Millisekunde hat, stellen wir das System darauf ein. Das ist das Grundkonzept – also im Prinzip ein Timer-Interrupt, der das Ganze antreibt, dann aber so wie jedes andere Echtzeitbetriebssystem funktioniert.

Das Besondere an unserem Echtzeitbetriebssystem beziehungsweise das, was uns von einem „normalen“ Echtzeitbetriebssystem unterscheidet, ist, dass wir mit Timern und zyklisch denken. Wenn man Echtzeitbetriebssysteme in anderen Embedded-Anwendungen betrachtet, stehen Interrupts und Events im Vordergrund, die echtzeitfähig verarbeitet werden. Das ist aber nicht das Konzept von PC-based Control: Das basiert auf einer zentralen, sehr performanten CPU. Dabei werden auch die neuesten Entwicklungen von Intel unterstützt, und je mehr Cores zur Verfügung stehen, desto höher ist die Performance der CPU. In Kombination mit einem schnellen Feldbus, wie zum Beispiel EtherCAT, automatisieren wir dann eine Maschine. Das Konzept ist recht einfach und auch deterministisch: So können wir genau sagen, wie viel Bandbreite auf dem Feldbus gebraucht wird, zu welchem Zeitpunkt die Eingänge eingelesen werden und wann sie wieder im PC verfügbar sind.

**Bringen neue Intel-Prozessoren Unruhe in die Beckhoff-Entwicklungsabteilung?**

Nein. Intel-CPU's sind ja auch deshalb so mächtig, weil sie immer abwärtskompatibel bleiben. Das heißt, wir können tatsächlich – natürlich ohne die neuesten Features der CPU zu nutzen – den TwinCAT-Programm-Code von 1996 auf der modernsten Intel-CPU ablaufen lassen. Ein anderer Fall ist es, wenn Prozessoren über eine ganz neue Architektur verfügen, wie bei der Multicore-Generation, für die auch das Betriebssystem angepasst werden muss. Für gewöhnlich sind dann auch die Verwaltungsmechanismen geändert worden. Hier müssen wir unsere Software entsprechend anpassen, oder es gibt noch einmal eine neue Entwicklung. Die aktuelle Prozessorarchitektur von Intel heißt Ivy-Bridge, davor war es Sandy-Bridge. Diese Weiterentwicklung brachte eine höhere Rechengeschwindigkeit, weil bei gleicher Taktfrequenz die Befehle schneller ausgeführt werden

und die Technologie insgesamt effizienter ist. Aber das erzeugt in unserer Entwicklungsabteilung keine Unruhe.

**War die Einführung der Multi-Core-Technologie der größte Entwicklungsschritt?** Das kann man so sagen. Wir haben lange Zeit auf Single-Core-Technologie gesetzt, wie auch Intel selbst. Intels Marketingstrategie war lange Zeit auf nur einen Core mit immer höherer Frequenz ausgerichtet. Das ließ sich hardwaretechnisch und physikalisch allerdings nicht durchhalten. Zu hohe Verlustleistung und Leckströme der damaligen Prozessoren ebneten den Weg zur modernen Multi-Core-Technologie. Das verlangte von uns natürlich eine komplette Neuentwicklung unserer Software-Architektur, denn ein einzelner Task kann nicht auf zwei Cores aufgeteilt werden. Da müssen schon vom Compiler her einzelne Bereiche parallelisiert werden. Das ist heutzutage mit modernen Compilern möglich, aber das geht zu Lasten der Überschaubarkeit und Deterministik: Wenn das System ein Eigenleben entwickelt, um mit maximaler Performance auf zwei oder mehr Cores



▲ (li.) Die Automatisierungssoftware TwinCAT 3.1 integriert zahlreiche neue Eigenschaften und Erweiterungen in Engineering und Runtime.

► (re.) Eingebettet in Microsoft Visual Studio bietet TwinCAT 3 die Möglichkeit, neben den Sprachen der IEC 61131 auch C/C++ und Matlab/Simulink zu nutzen, um Applikationen in Echtzeit zu programmieren und zu debuggen.

zu laufen, ist es schwierig die Kontrolle darüber zu behalten, wie schnell etwas tatsächlich abläuft. Deswegen haben wir bei der Erweiterung von TwinCAT auf Multi-Core den Ansatz verfolgt, dem Scheduler beizubringen, mit mehreren Cores zu arbeiten und die Tasks dann, je nach Konfiguration, auf einem bestimmten Core ablaufen zu lassen.

### Welche Möglichkeiten eröffnet Multi-Core Ihren Kunden?

Primär geht es um die höhere Systemperformance. Die Leistungssteigerung entsteht, wenn zum Beispiel die Motion-Control-Tasks von den PLC-Tasks getrennt werden und auf zwei Cores ablaufen. Die Tasks werden dann physikalisch parallel verarbeitet. Wir haben die einzelnen Tasks von Anfang an als Module gekapselt, da wir bei TwinCAT schon immer an das Software-Volumen gedacht haben.

In unserem Entwicklungslabor testen wir aktuell ein 32-Core-System, das mit einem 64-Bit-Betriebssystem läuft – und das Ganze in Echtzeit. Da sind anspruchsvolle Software-Module gefragt, um die gebotene Leistung auszureizen. Unser Hauptaugenmerk liegt jetzt darauf, Funktionen zu erzeugen, die für den Anwender greifbar sind. Ein Thema ist, eine Sicherheits-SPS in Software zu entwickeln, die auf einem Core läuft. Aber auch unser eXtended Transport System oder ein Condition-Monitoring-Modul, das sehr rechenzeitintensiv ist, könnten auf einem gesonderten Core ablaufen, ohne den übrigen Prozessor zu belasten. Wir werden Funktion nach Funktion erzeugen, um diese hohe Systemperformance sinnvoll zu nutzen und damit unseren Kunden den eigentlichen Mehrwert zu liefern. In diesem Zusammenhang ist auch das Thema Zykluszeit sehr wichtig, weil wir mit Multi-Core-Technologie die Zykluszeit weiter reduzieren können. Für den Endkunden heißt das, dass er größere Maschinen mit kürzeren Zykluszeiten betreiben kann.

Ein weiterer Aspekt ist die Kostenreduzierung des Gesamtsystems: In großen Anwendungen können verteilte Steuerungen in ei-



ner einzigen, sehr performanten Steuerung zusammengefasst werden. Das reduziert Wartungskosten und senkt die TCO.

**Welche Neuerungen bietet TwinCAT 3.1?** Eines der Keyfeatures der neuen TwinCAT-Version wird die 64-Bit-Fähigkeit sein. Das bringt noch einmal eine deutliche Performancesteigerung und erlaubt zudem die Adressierung von mehr Speicher. Erste Benchmarks haben ergeben, dass wir, bei gleicher Taktfrequenz, bei der Berechnung von normalen Instruktionen circa 20 Prozent schneller sind. Auf der Hannover Messe 2013 werden wir die ersten Benchmarks zeigen, damit unsere Kunden sehen, was die 64-Bit-Technik bringt. Die interessanteste Information ist dabei, dass die aktuellen CPUs alle schon 64-Bit-fähig sind und dass mit einem 64-Bit-Betriebssystem, wie Windows 8, das ganze TwinCAT-System durchgängig mit 64 Bit arbeitet.

Darüber hinaus ist ein wesentlicher Entwicklungsschritt, an dem wir gegenwärtig arbeiten, die Zykluszeiten noch weiter zu reduzieren. Sprachen wir bis eben noch von 100 µs, testen wir heute schon Systeme in unserem Labor, die mit 12,5 µs arbeiten; und selbst diesen Zyklus werden wir noch weiter verringern. – Immer genauer und immer schneller zu werden, das ist unser Ziel.

Die Fragen stellte Peter Schäfer.

► [www.beckhoff.de](http://www.beckhoff.de)