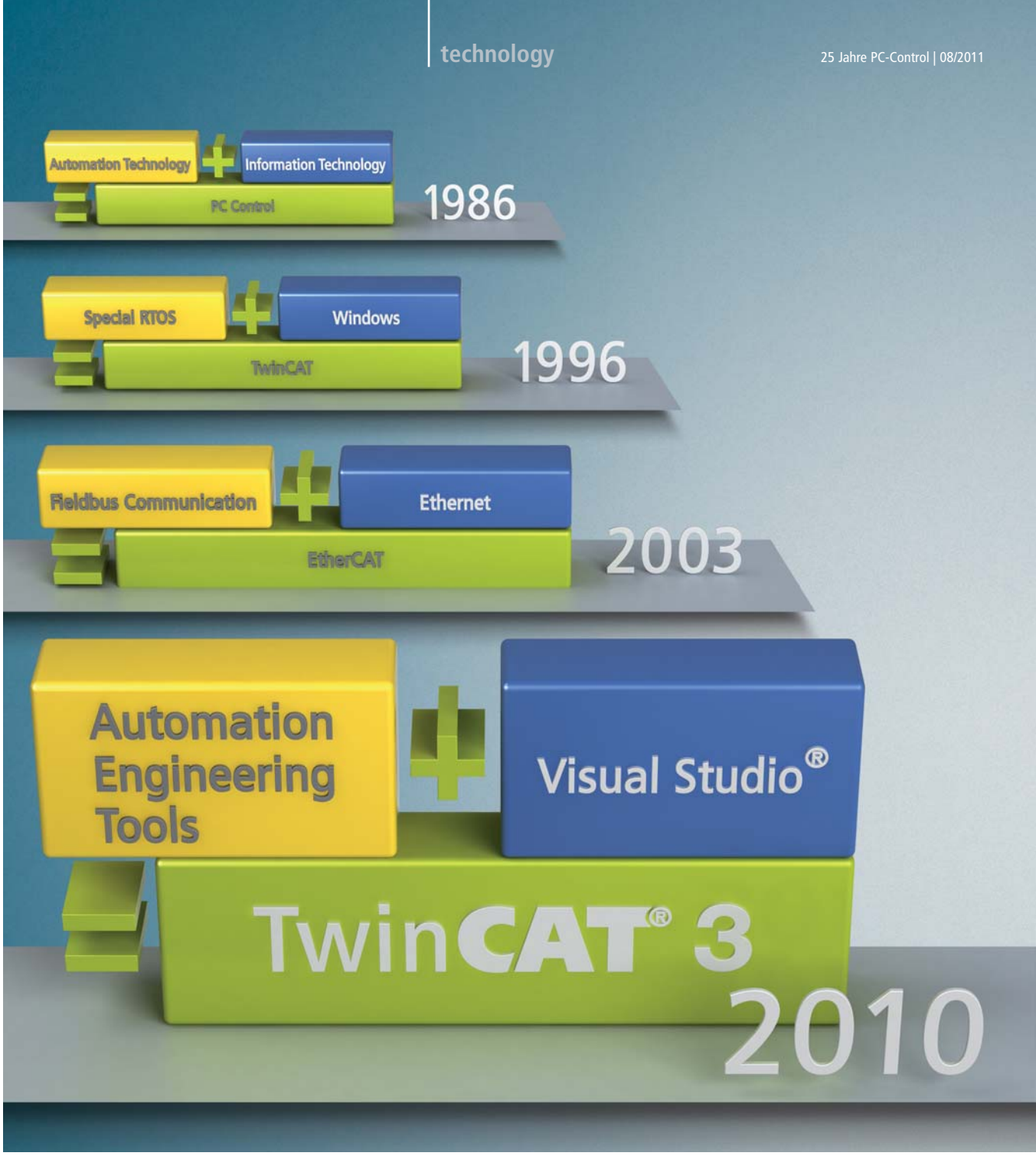


eXtended Automation Engineering

TwinCAT 3: C++ drives Automation

Mit TwinCAT 3, der neuesten Softwaregeneration von Beckhoff, können SPS-Programmierer, neben den traditionellen IEC 61131-3-Sprachen, nun auch C/C++ zur komfortablen Programmierung nutzen. C++ als Ablaufsprache in der Beckhoff-Steuerung ist allerdings nicht neu: Schon die DOS-basierte PC-Steuerung S2000, von Beckhoff im Jahr 1993 entwickelt, konnte C als Ablaufsprache einbinden.

Mit der S2000 brachte Beckhoff ein auf Microsoft-DOS basierendes, durch reine Software-Erweiterungen echtzeitfähiges Steuerungssystem auf den Markt, das auf einer einzigen CPU die Abarbeitung von SPS, Motion und Visualisierung erlaubte. Als Single-Task-Betriebssystem konnte DOS jedoch immer nur ein Programm starten. Sollte ein anderes Programm zum Ablauf gebracht werden, musste das aktuell bearbeitete zuerst beendet werden. Um dennoch Hintergrundfunktionen zu ermöglichen und nach dem Beenden des Programms kleine Hilfsprogramme im Speicher zu belassen, nutzte man TSR (Terminate Stay Resident), das durch Interrupts aktiviert wurde. Die S2000-Software unterschied drei Taskebenen: die NC-Task mit höchster Priorität, gefolgt von der SPS-Task und der Vordergrund-Task, der Bedienoberfläche, die in der restlichen verfügbaren Rechenzeit des PCs ablief. Die Umschaltung zwischen den Taskebenen erfolgte durch einen Scheduler. Die S2000-Vordergrund-Task beinhaltete ein kooperatives Multitasking, das Benutzereingaben parallel zu Hintergrundfunktionen ermöglichte, womit die Singletask-Einschränkung des DOS-Betriebssystems behoben war. Die visuelle Darstellung der Bedienoberfläche erfolgte über textuelle Grafik mit 80 x 25 Zeichen und ASCII-Zeichensatz. Als Hintergrund-Task konnte z. B. mit C++ die Kommunikation zu einer Datenbank aufgebaut werden, um neue Maschinenaufträge zu erhalten oder einen produzierten Auftrag zu quittieren.

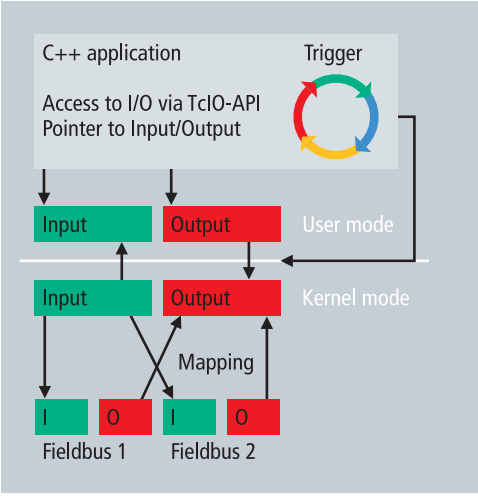
Begrenzte Hardware-Ressourcen: eine der größten Herausforderungen

Eine der größten Herausforderung für die C++-Entwickler bestand im Engineering auf einer Plattform mit sehr begrenzter Hardware-Ressource: DOS stellte nur einen 640-kB-Arbeitsspeicher zur Verfügung; darin musste die Applikation mit Motion, SPS, HMI und eigenem C++-Code ablaufen. Hilfstools, wie QEMM, bescherten Glücksmomente, da mit dem „Quarterdeck Expanded Memory Manager" der verfügbare Speicherbereich oberhalb der 640-kB-Grenze etwas erhöht werden konnte. Trotzdem war es fast nie möglich, die gesamte Applikation im neu erstellten C++-Code zum Debuggen und mit Debug-Informationen in den Speicher zu laden. Das Auffinden von Fehlern war sozusagen eine sportliche Aufgabe, da immer nur wenige Teilmodule mit Debug-Information in dem begrenzten Speicher ablaufen konnten. Heutige TwinCAT-C++-Programmierer schmunzeln über Ressourcen- bzw. Speicherbegrenzungen, dabei gehört dies gegenwärtig für viele Entwickler auf Nicht-Beckhoff-Embedded-Lösungen immer noch zum Tagesk(r)ampf: Fehlende Online-Monitoring-, Diagnose- oder Debug-Möglichkeiten werden mit der Ausgabe von Testsignalen auf einem seriellen Port bekämpft: „Welcome to the history!"

TwinCAT CE bietet Feldbus-Infrastruktur für C++

Mit Windows CE hat Microsoft ein zum Desktop-Windows kompatibles Betriebssystem entwickelt, das in Größe und Funktion fein skaliert werden kann. Ursprünglich vor allem für den Einsatz in mobilen Geräten geplant, ist Windows CE seit der Version 3.0 echtzeitfähig und wird vermehrt im industriellen Umfeld eingesetzt. Aufgrund der Verwandtschaft von Windows CE mit dem „großen" Windows konnte „TwinCAT CE" Quellcode-kompatibel zu „TwinCAT XP" angepasst werden. Da TwinCAT alle Echtzeitfunktionen auf Windows CE abgebildet hat, können TwinCAT-Echtzeitanwendungen, wie z. B. Software-SPS und -Motion-Control, mit Echtzeitanwendungen anderer Herkunft koexistieren. TwinCAT CE bietet neben den SPS- und Motion-Funktionalitäten aber auch die Möglichkeit, aus eigenem, nativen C-Code, einen bis in den Feldbus durchsynchronisierten, deterministischen Zyklus zu fahren. Das TwinCAT-System bietet dabei die Flexibilität, die physikalischen Ein- und Ausgänge von den unterschiedlichsten Feldbussystemen zunächst in logische I/O-Tasks zu mappen. Die Pointer auf die logischen Ein- und Ausgangsprozessabbilder stehen in der C-Applikation zur Verfügung, ebenso wie der TwinCAT-Echtzeit-Callback. Aus diesem deterministischen, zyklischen Callback kann zunächst mit einer Methode das Einlesen der physikalischen Eingänge angestoßen werden, bevor nach der eigentlichen Logikberechnung das Verteilen der logischen Ausgänge auf die physikalischen Ausgänge ausgeführt wird.

C-Anwender schätzen an der TwinCAT-CE-Lösung die Nutzung einer offenen Automatisierungsplattform, die es ermöglicht, den bisher auf anderen Betriebssystemen lauffähigen C-Code, mit minimalen Anpassungen, weiterhin als zyklischen Logik-Code nutzen zu können: Weder die Anbindungen an verschiedene Feldbussysteme noch die Connectivity in die HMI- und ERP-Welt müssen extra implementiert werden – sie sind bereits in der Infrastruktur des TwinCAT-CE-Systems enthalten.



TwinCAT-2-CE-Geräte ermöglichen den Ablauf von deterministischem C-Code bis in die verschiedenen Feldbussysteme.

Stefan Hoppe, Produktmanager TwinCAT, Beckhoff Automation

TwinCAT 3 C++: Extended

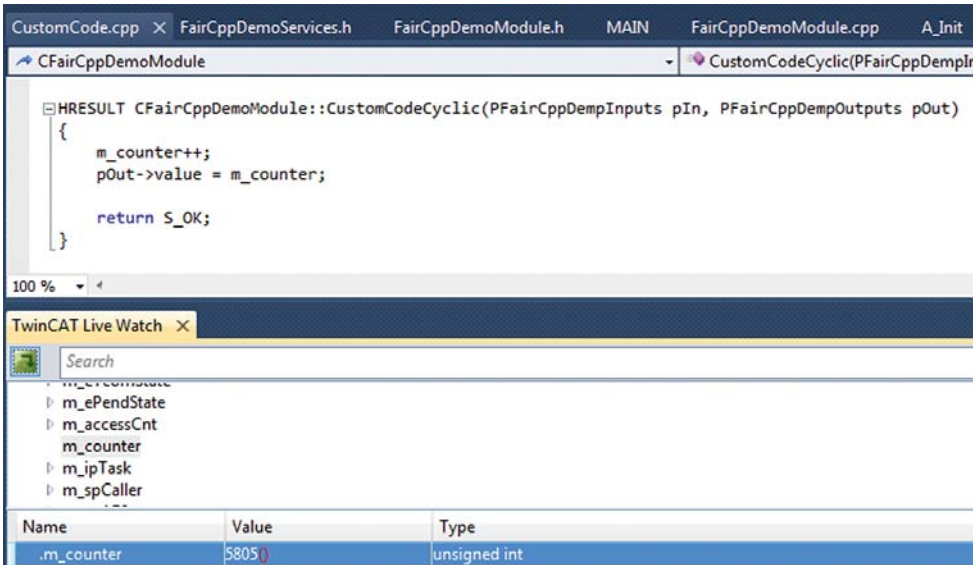
C++ mit TwinCAT 3 bedeutet für Entwickler zunächst, dass mit Microsoft Visual Studio® 2010 ein leistungsfähiger, moderner Editor zur Verfügung steht: Die Source-Safe- oder Team-Foundation-Anbindung zur Sicherung und Versionierung von Code ist nur ein Vorteil der neuen Engineering-Umgebung, welche vor allem die Arbeit im Team untereinander erleichtert. Schnelleres und effektives Engineering von C-Code klingt nach purem Marketing – faktisch bietet aber die TwinCAT-3-Automatisierungsplattform dem C-Programmierer nie gekannte Möglichkeiten und Freiheiten: C-Code kann zyklisch ablaufen, um auf die physikalische I/O-Ebene zuzugreifen, oder auch nur mathematische Funktionen bereitstellen, die On-Demand, von anderen C-Modulen oder auch vom SPS-Programmierer, aufgerufen werden können. Auf der Basis von Multi-Core-CPUs können, durch einfaches Konfigurieren, nun mehrere Instanzen vom erstellten C-Modul leicht auf verschiedene CPU-Cores verteilt werden. Auch hier stehen alle Basisfunktionalitäten zur Verfügung: Sollen im C-Code Aktionen gestartet werden, die eine längere Bearbeitung benötigen als der zyklische Takt erlaubt, so müssen diese Aufgaben entkoppelt werden. Beckhoff stellt dafür ein „Software Development Kit" (SDK) zur Verfügung, um Aktionen aus dem deterministischen Echtzeitablauf starten und den Bearbeitungsstatus überwachen zu können. Das Einlesen bzw. Schreiben von Dateien, das Starten von Threads, das Allokieren von Speicher und das Kommunizieren mit Datenbanken erfolgen somit über Funktionen aus dem SDK und entsprechen damit dem, für SPS-Programmierer bekannten, Mechanismus der Verwendung von SPS-Libraries. Der in Microsoft Visual Studio® 2010 enthaltene C-Compiler wird für die C-Code-Generierung genutzt. Der Zielcode wird nach der Kompilierung, in Form von dynamisch ladbaren Bibliotheken (DLL), in das Echtzeitlaufzeitsystem geladen. Der Anwender ist somit in der Lage, das TwinCAT-System durch Programmmodule zu erweitern. Diagnosemöglichkeiten, wie das Monitoring und das Verändern von Prozessvariablen bei laufendem SPS-Zyklus, sind für SPS-Programmierer eine Selbstverständlichkeit.

Ein C-Entwickler muss mit dem Setzen eines Breakpoints zunächst den Ablauf des Codes anhalten, um in Watch-Fenstern den Status der Symbole zu modifizieren. Auch hier hat Beckhoff die Möglichkeiten des Visual Studio® erweitert und für den Einsatz von C/C++ in der Automatisierungstechnik optimale Diagnosemöglichkeiten geschaffen: Mit Hilfe des „Custom Debugger Interface" wurde ein eigener Debug-Transportkanal geschaffen. Die Onlinedaten stehen anschließend in einem „TwinCAT-Watch"-Fenster zum Monitoring und für Veränderungen zur Verfügung, und zwar ohne den Maschinenablauf per Breakpoint stoppen zu müssen. Diese Diagnose der Echtzeitanwendung steht sowohl lokal als auch über das Netzwerk zur Verfügung.

Willkommen in der Zukunft!

C++ in der Automatisierungstechnik ist nicht neu; mit TwinCAT 3 steht aber ein neues, leistungsfähiges Modulkonzept zur Verfügung, welches - in Kombination mit den skalierbaren Multi-Core-CPU-Möglichkeiten und dem durch Beckhoff erweiterten Visual-Studio®-C++-Monitoring ein viel flexibleres, schnelles Engineering ermöglicht: „Welcome to the future!".

www.beckhoff.de/TwinCAT3



Das Monitoring von C++-Variablen kann unter TwinCAT 3 mit oder ohne Breakpoints erfolgen.