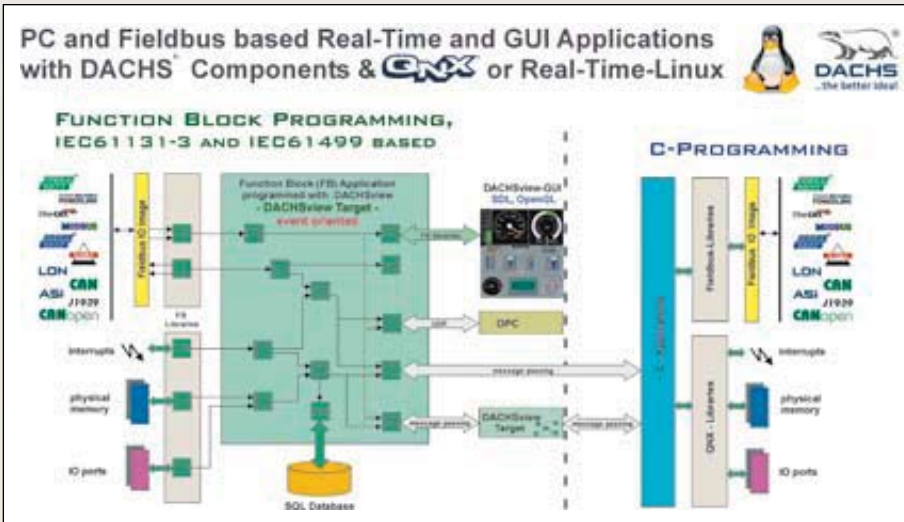


Real-Time für den Maschinenbau

An die Steuerungstechnik von hochwertigen Maschinen und Anlagen werden immer höhere Anforderungen gestellt. Laut einer Tendenzbefragung des VDMA1 sind immer öfter verteilte Systeme gefragt, bei denen auch Steuerungs-, I/O- und Grafikfunktionen im System integriert sind.



Natürlich sind verteilte Systeme oder ereignisorientierte Echtzeitverarbeitung nicht mit dem 17 Jahre alten IEC 61131-3-Standard möglich. An neuen Wegen führt daher nichts vorbei, will man aktuellen Anforderungen gerecht werden.

Mit der neuen Version 1.3 von DACHSview für QNX 6.4 und PREEMPT_RT LINUX bietet Steinhoff Automation & Fieldbus-Systems zwei Multi-Core Soft-SPS RTOS-Versionen an, die genau die oben geforderten Eigenschaften abdecken, wobei im Design-Konzept auch auf die Minimierung der Systemkomplexität geachtet wurde. Im Folgenden wird aufgezeigt welche Probleme existierten und wie diese bei der Implementierung von DACHSview elegant gelöst wurden.

Sind PC-basierte Real-Time Betriebssysteme die Lösung?

Bei Real-Time-Betriebssystemen (RTOS) denkt man zuerst an den Einsatz von komplexen Entwicklungsumgebungen mit textuellen Sprachen, für die in der Regel Real-Time-Experten als Entwickler benötigt werden. Ein solcher Ansatz ist sehr kostenintensiv, wobei wegen der hohen Anfangsinvestitionen auch noch eine starke Abhängigkeit zum RTOS-Hersteller entsteht. Ein weiterer kritischer Punkt von PC-basierten Systemen, auch Programmable-Automation-Controller (PAC) genannt, ist die Komplexität des gesamten Systems mit Handlings-, Installations- und Programmier-Problemen. Die vielen internen Schnittstellen, mit denen der Softwareentwickler konfrontiert wird, sind auch nicht zu

unterschätzen. Folglich sehen PC-basierte Real-Time-Betriebssysteme auf den ersten Blick ungeeignet aus, käme man nicht auf die Idee, RTOS-Applikationen in einer universellen grafischen Programmiersprache zu entwickeln, anstatt eine textuelle Sprachen zu verwenden.

Grafische Programmiersprachen für Real-Time-Applikationen

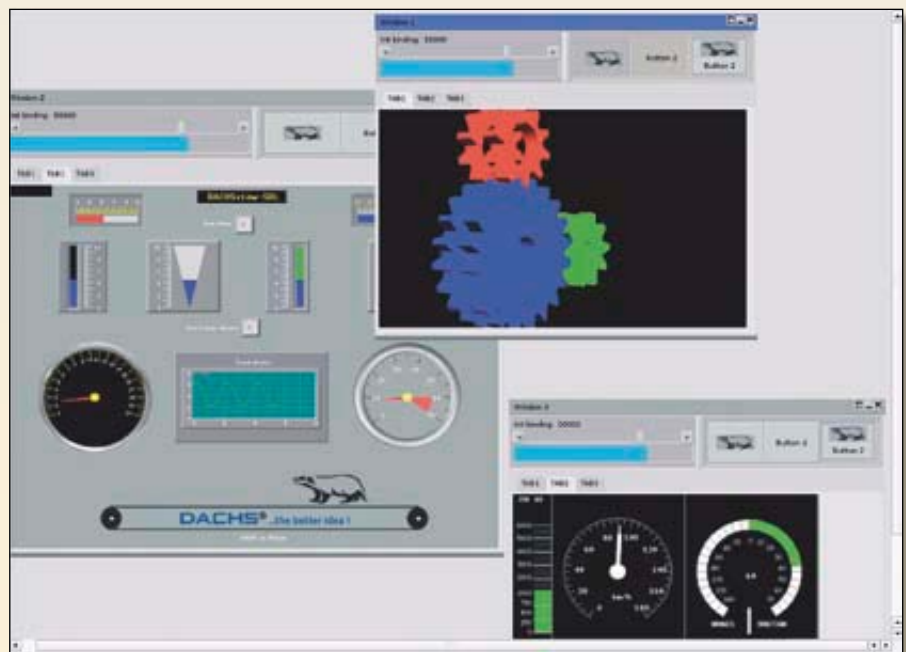
Textuelle Sprachen, wie C und C++, können in die Steuerungswelt nicht ohne Akzeptanzverluste übertragen werden. Ein naheliegender Ansatz zur Lösung dieses Dilemmas ist der Ersatz der textuellen Sprachen

durch eine universelle grafische Programmiersprache, die als eine allgemeine Entwicklungssprache für Real-Time- bzw. Steuerungssysteme eingesetzt wird.

Nach den positiven Erfahrungen mit den grafischen Sprachen des IEC 61131-3-Standards liegt es nahe eine zu den textuellen Sprachen äquivalente grafische Sprache einzusetzen. Hier bietet sich die bekannte Funktionsblockdarstellung (FBD) an, da sie als formale Sprache im Vergleich zu den Textsprachen gleichmächtig ist. Mit dem SPS-Standard IEC 61131-3 wurden jedoch in der FBD-Restriktionen eingeführt, die erst aufgehoben werden müssen, um funktional mit den Textsprachen gleichzuziehen. Es müssen beispielsweise FBD-Programme vom RTOS als normale RTOS-Applikationen behandelt werden, d.h. sie müssen unter der Kontrolle des RTOS-Schedulers ausgeführt werden. Dies ermöglicht, dass blockierende Funktionen unterstützt werden können, die auf RTOS-Ereignisse suspendiert warten und nicht aktiv zur Ereigniserkennung pollen müssen. Nur so können Reaktionszeiten im Bereich von Mikrosekunden erreicht werden, womit eine wesentliche Grundlage für die Echtzeitverarbeitung gegeben ist.

FBD als „general purpose“ Entwicklungssprache

Mit Funktionsblöcken sind unterschiedlichste Systemdetails kapselbar, womit der





erste Schritt zur Minimierung der Komplexität getan werden kann. Mit der Kapselfung wird nebenbei noch eine Entkopplung von den verwendeten Softwaretechnologien erreicht. Funktionen von Teilsystemen, wie Grafik-, Feldbus-, Hardware- und Kommunikationsfunktionen werden einheitlich über FBD-Bibliotheken implementiert. Diese Funktionen sind somit für den Entwickler nahtlos ohne Schnittstellenprobleme per Mausklick nutzbar. Durch die Möglichkeit eigene, hierarchische Funktionsblöcke zu definieren und diese zu instanzieren oder wiederzuverwenden, können intuitiv objektorientierte Konzepte auch von Mitarbeitern realisiert werden, die nicht unbedingt Informatiker sein müssen. Die FBD wird so im Sinne der objektorientierten „general purpose“ Entwicklungssprache für Real-Time-Applikationen und -Steuerungen eingesetzt. Durch diese freie Nutzung der FBD wird die Komplexität der Softwareentwicklung auf ein Minimum gebracht.

Auch das Laufzeitsystem der Real-Time-SPS musste funktional erweitert werden. Im Gegensatz zur IEC 61131-3 basierten SPS kann eine Real-Time-Soft-SPS für modulare Applikationen mehrfach auf einem Zielsystem gestartet werden, womit gekapselte modulare Applikationen möglich sind. Jede Real-Time-Soft-SPS kann mehrere Steuerungsthreads als Systemthreads starten. Wichtig ist, dass der Entwickler der Steuerungsapplikation keinen direkten Kontakt mit dem RTOS des Zielsystems benötigt.

Die FBD-Programmierung geschieht bei DACHSview unter MS-Windows auf einer PC-Workbench, die über eine UDP-Verbindung den Upload der Real-Time-Applikation zur remoten Real-Time-SoftSPS DACHSview-RT und den Online-Test unterstützt.

FBD-Bibliotheken

FBD-Bibliotheken gibt es für Schnittstellen von Hardware, Feldbussen, Grafik,

Datenbank, mathematische Funktionen usw. DACHSview stellt dem Steuerungsprogrammierer FBD-Bibliotheken zur Verfügung mit denen er Betriebssystemfunktionen bis runter zum Durchgriff auf Hardware nutzen kann. Des Weiteren existieren FBD-Bibliotheken für statische und animierte 2D- und 3D-OpenGL-Grafiken. Diese Grafikfunktionen erlauben die Programmierung einzelner Pixel bis hin zur Nutzung komplexer Widgets.

Für die I/O-Ebene werden FBD-Bibliotheken für Ethernet basierte Feldbusse wie Profinet (RT/IRT), EtherCAT, Ethernet Powerlink und Modbus angeboten. Auch die klassischen Feldbusse Profibus, Interbus, CAN und CANopen werden unterstützt. Über diese Bibliotheken ist die nahtlose Integration von Feldbusdaten mit anderen Systemfunktionen möglich.

Mathematische Funktionen aus dem ROOT/CINT-System vom CERN2 können speziell für die auf Real-Time-Linux basierte Version genutzt werden. Dieses System mit dem C/C++ Interpreter CINT stellt den Forschern vom CERN interessante Datenvisualisierungen und mächtige mathematische Bibliotheken für Datenauswertungen und Statistiken zur Verfügung.

Lokale und verteilte Systeme kommunizieren über Message-Passing-Middleware. Lokale und verteilte PC-Systeme werden mit DACHSview über die Message-Passing-Middleware ZeroMQ unterstützt, die eine sichere und mikrosekundenschnelle Kommunikation zwischen Linux, QNX Neutrino, MS-Windows und anderen Betriebssystemen bietet. Dies geschieht natürlich über eine FBD-Bibliothek.

Embedded-PC-Hardware als Steuerungsplattform

Als Hardwarebasis steht heute PC-Hardware mit sehr leistungsfähigen Low-Power-CPU's zur Verfügung, die fallweise mehrere CPU-Cores besitzen. Diese leistungs-

fähigen, lüfterlosen Embedded-PC-Systeme bieten die ideale Basis für die heute geforderte Systemintegration. Da man die unterschiedlichen Systemprozesse und -Threads einzelnen CPU-Cores zuordnen kann, sorgen Multi-Core-CPU's dafür, dass sicherheitskritische Funktionen die geforderte Rechenzeit bekommen können.

Zusammen mit diesen Hardwarevoraussetzungen bieten heute ausgereifte PC-basierte Real-Time-Betriebssysteme, wie das preemptive Real-Time Linux oder QNX Neutrino, alle Eigenschaften, um die geforderte Systemintegration und das erforderliche Echtzeitverhalten zu leisten.

Komplettlösungen

Um dem Endbenutzer den Handlings-, Installations- und Programmier-Aufwand für die Integration aller Hardware- und Softwarekomponenten zu ersparen, wird von Steinhoff ein komplettes System samt vorinstalliertem RTOS, der Real-Time-SPS-, Feldbus-Hard- und Software sowie konfigurationsspezifischer Module angeboten. Das System kann wahlweise unter dem preemptiven Real-Time Linux oder QNX Neutrino (ab Vers. 6.4.1) betrieben werden. Passend dazu wird die DACHSview-Entwicklungsumgebung für MS-Windows kundenspezifisch mit den optionalen RTOS-, Feldbus- und Grafikbibliotheken usw. angeboten.

Fazit

Zusammenfassend ist festzustellen, dass die freie Verwendung der Funktionsblockdarstellung für alle Systemaspekte zu einer echten Innovation führt, die Kompliziertes einfacher macht. Das Resultat ist eine durchgängige System- und Entwicklungsumgebung, die durch die Integration von Grafik-, SPS-, Feldbus-I/O-, Datenbank- und Kommunikationsfunktionen das effiziente Entwickeln von umfassenden Steuerungsapplikationen kostengünstig ermöglicht.

Damit erfüllt Steinhoff nicht nur die Technikrends entsprechend der Tendenzbefragung des VDMA mit verteilten Systemen mit integrierten Steuerungs-, I/O- und Grafikfunktionen, sondern kann zusätzlich neue Technikrends jederzeit über Funktionsblockbibliotheken integrieren.

DACHSview wird international eingesetzt, u.a. in Materialflusssystemen, wie beispielsweise automatischen Parkhäusern. Seine grafischen Fähigkeiten hat DACHSview besonders bei der Entwicklung von ICE-Lokführerdisplays, basierend auf dem ETCS-Standard, bewiesen.

■ STEINHOFF

Automation & Fieldbus-Systems
www.steinhoff-automation.com
info@steinhoff-automation.com