

dem gleichen magnetischen Impulsfeld von 3 kA/m ausgesetzt. Hierbei wird die induzierte Spannung U_2 in der ungeschirmten Leitung gemessen. Aus den Messwerten ergibt sich die magnetische Schirmdämpfung nach der Formel: $\alpha_s = 20 \log(U_1/U_2)$ dB.

Durch die Versuche und Simulation mit einem FEM-Programm (FEM, Finite Elemente Methode) konnte die magnetische Schirmwirkung α_s eines Kabeltrag-Systems eindeutig nachgewiesen werden. Das beste Ergebnis von rund 50 dB wurde bei Kabeltrag-Systemen (Kabelinnen) mit Deckel erzielt (Tafel 1).

Transferimpedanz von Kabeltrag-Systemen

Die Transferimpedanz (Kopplungswiderstand) eines Kabeltragsystems ist das Verhältnis von gemessener Spannung $U_{\text{Stör}}$, die in Längsrichtung innerhalb des Kabeltragsystems gemessen wird, zu dem eingekoppelten Strom $I_{\text{Stör}}$. Die Transferimpedanz wird in Analogie zur Messung der elektrischen Leiteigenschaften nach Kapitel 11.1. (DIN EN 61537) bestimmt.

Bei einem Blitzeinschlag in ein Gebäude fließen Blitzteilströme im gesamten Potentialausgleichssystem. Da installierte Kabel und Leitungen vorteilhaft innerhalb eines Kabeltrag-Systems verlegt werden und installierte Kabeltrag-Systeme immer mit in das Potentialausgleichssystem einbezogen sind, fließt dabei der Blitzteilstrom über das Kabeltrag-System. Ein sehr kleiner Anteil kann daher noch über die innerhalb des Kabeltrag-Systems verlegten Leitungen fließen. Dieser Anteil wird durch die Transferimpedanz des Kabeltrag-Systems bestimmt. Für die Transferimpedanz gilt: $Z_T = U_{\text{Stör}}/(I_{\text{Stör}} \times l)$ [mΩ/m (Milliohm/Meter)]. Die angegebenen Werte basieren auf Messungen, bei denen ein Impulsstrom der Wellenform 8/20 μs durch eine definierte Länge eines Kabeltrag-Systems eingespeist wurde.

Die Wirkung des Kabeltrag-Systems gegen galvanische Kopplung wurde durch die Versuche ebenfalls eindeutig nachgewiesen. Das beste Ergebnis wurde bei Systemen mit Deckel erzielt (Tafel 1).

K. Jendralski

Windows als Echtzeit-Betriebssystem

Die Anforderungen an Steuerungen in der Automatisierung steigen stetig. Neben klassischen Aufgaben wird immer häufiger auch eine Verbindung mit Robotik, komplexen mathematischen Berechnungen oder Bildverarbeitung benötigt oder die Nutzerinteraktion soll ein modernes Erscheinungsbild tragen. Hier zeigt sich vor allem der Vorteil PC-basierter Lösungen. Dazu ist auch das neue Windows 7 gut geeignet, wenn ihm eine leistungsfähige Echtzeiterweiterung zur Seite gestellt wird.

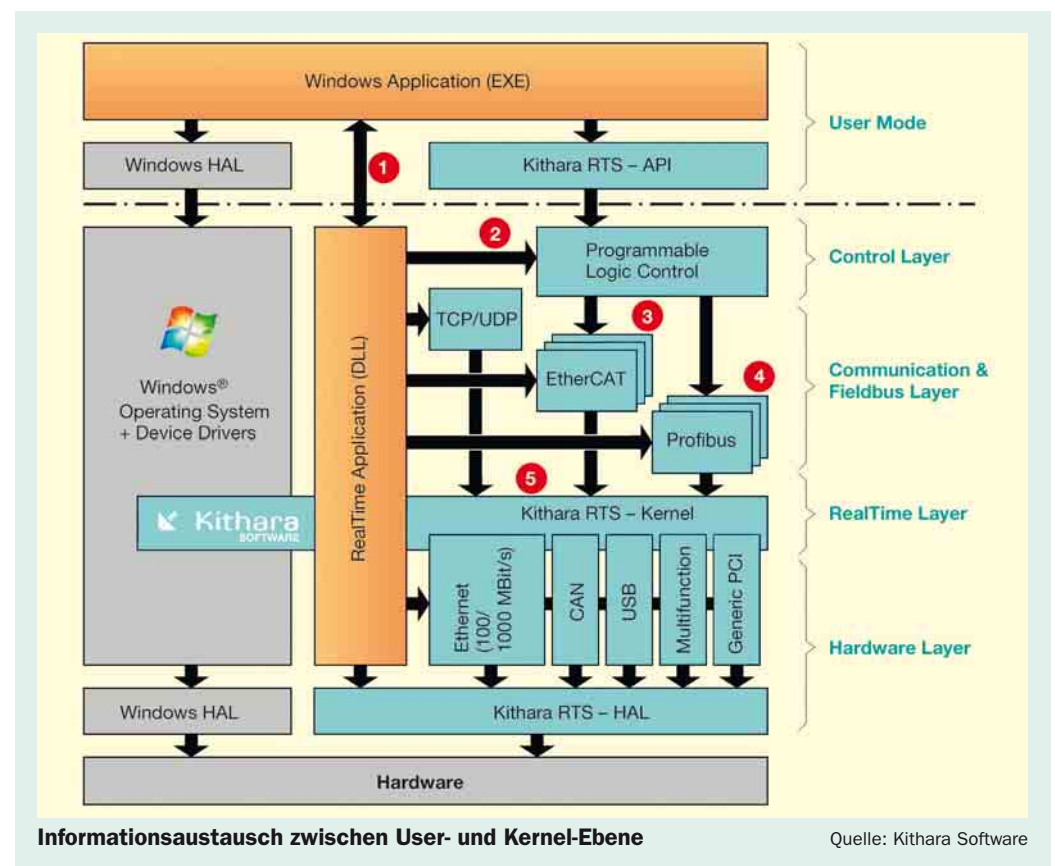
Endkundentechnologien in der Industrie

Viele in Consumer-Geräten angewandte Technologien werden anfänglich für nicht industrietauglich oder zumindest als dort weniger geeignet eingeschätzt. Nach einiger Zeit, wenn sich die Technologie aufgrund der immens hohen Stückzahlen in der Consumerbranche weit verbreitet hat, lässt sich beobachten, dass diese doch noch Einzug in industrielle Anwendungen hält. Die Gründe dafür sind vor allem darin zu suchen, dass aufgrund der hohen Stückzahlen die Produktion sehr kostengünstig und das Know-How der Technologie breit verfü-

bar ist. Nach einiger Reife erfüllt die Technologie trotz der in der Regel strengen Anforderungen seitens der Industrie dennoch die Kriterien für entsprechende Anwendungen.

So ist es nicht nur beispielsweise Ethernet als Kommunikationstechnik ergangen, das noch in den 90er Jahren des vergangenen Jahrhunderts als für industrielle Kommunikation, bei der oftmals Echtzeit gefordert wird, gänzlich ungeeignet erschien. Im Gegensatz dazu gibt es heute bereits über 20 verschiedene auf Ethernet basierende Automatisierungsprotokolle und es ist aus der Automatisierung nicht mehr wegzudenken.

Ein weiteres Beispiel für späteren Ruhm ist der Einsatz der PC-Technik im Allgemeinen und Windows als Betriebssystem im Besonderen im Rahmen von industriellen Applikationen. Dabei ist die Einschätzung, ob Windows überhaupt für Echtzeit-Aufgaben geeignet sei, sehr unterschiedlich. Einerseits sind die aktuellen PC-Prozessoren so leistungsfähig, dass man meinen könnte, Windows selbst würde schon Aufgaben in Echtzeit bearbeiten können, was aber allenfalls als „Schön-Wetter-Echtzeit“ zu bezeichnen wäre, denn der Begriff schließt bekanntlich alle Situationen ein, auch und gerade die unerwünschten. Allgemein anerkannt ist daher, dass man selbstverständlich eine spezielle Echtzeiterweiterung benötigt, um unter Windows zeitkritische industrielle Anwendungen realisieren zu können. Andererseits hält sich dennoch bis heute mitunter eine generelle Skepsis, ob denn selbst eine Kombination aus Windows als Standard-Betriebssystem und einem Echtzeit-Unterbau ausreichend sein würde. Dies kann – und das untermauern auch die seit vielen Jahren realisierten Lösungen auf dieser



Basis – heute als erwiesen angesehen werden. Auf Windows basierende Echtzeitleösungen sind in vielfältigen Anwendungen im Einsatz und bieten eine Leistungsfähigkeit, die auch komplexe Aufgaben zu lösen gestattet. Im Folgenden wird sich auf die Echtzeiterweiterung RealTime Suite des Unternehmens Kithara Software bezogen, die in der Version 9.1 vorliegt und mehr als 20 verschiedene Module mit über 250 API-Funktionen enthält.

Echtzeiterweiterung für Windows

Um eine Echtzeiterweiterung für Windows einsetzen zu können, sind grundsätzlich mehrere Bedingungen zu erfüllen: Einerseits muss es eine Echtzeit-Umgebung geben, die die zugrunde liegende Hardware direkt anspricht, um die geforderten Eigenschaften für zeitkritische Aufgaben zu erreichen. Andererseits sollte der normale Betrieb von Windows nicht beeinflusst werden. Ausnahmen sind in der Regel nur kurzzeitig zulässig.

Bei der Echtzeiterweiterung RealTime Suite wird ein Echtzeit-Multitasking-System errichtet, welches Windows selbst als eine sehr niedrig priorisierte Task auffasst. Die Behandlung der zeitkritischen Interrupts von Timern und industrienspezifischen Kommunikationskarten wird direkt an der Interrupt-Controller-Hardware vorgenommen. Nur durch die Absicherung der höchsten Systempriorität wird die geforderte Echtzeitfähigkeit gewährleistet.

Um dem Anwender diese Zusage weiterzugeben, ist kein eigenständiges RTOS (Real-Time Operating System) aufzusetzen. Stattdessen wird ihm eine Möglichkeit geboten, seinen zeitkritischen und hardwarenahen Programmcode in Form einer speziellen DLL in den Echtzeit-Kontext zu verlagern. Der Vorteil besteht darin, dass die gewohnte Entwicklungsumgebung und Programmiersprache weiterhin verwendet werden kann und sich dadurch der Einarbeitungsaufwand in Grenzen hält. Bedingung ist, dass der verwendete Compiler nativen Maschinencode erzeugen muss. Dies gilt für verschiedene Entwicklungssysteme für C/C++

sowie die Delphi-Entwicklungsumgebung. Damit stellt sich unmittelbar die Frage, wie die .NET-Umgebung und damit beispielsweise die Sprache C# (C-Sharp) ebenfalls eine Basis für industrielle Anwendungen sein kann. Grundsätzlich wird auch eine Anbindung für C# bereitgestellt, so dass sämtliche Ressourcen der industriellen Applikation auch dort verwaltet werden können. Die zeitkritischen und hardwarenahen Teile lassen sich zum Beispiel in C++ programmieren und in eine DLL auslagern, die dann vom Echtzeitsystem direkt auf die Kernel-Ebene geladen wird.

Im Ergebnis ist auch unter Windows ein Echtzeit-Multitasking realisierbar, wie man dies von einem reinen RTOS erwartet. Es stehen insgesamt 255 Prioritätsstufen zur Verfügung. Die für industrielle Anwendungen unbedingt erforderliche Vorhersagbarkeit der Abläufe wird durch eine Prioritätsvererbung zur Vermeidung der Prioritätsinversion erreicht. Die Prioritäten der Tasks lassen sich auch dynamisch anpassen. Es stehen Semaphore und Events zur Synchronisation der Tasks bereit.

Die Kommunikation des Echtzeit-Teils mit der übrigen Anwendung ist einfach und direkt möglich, da durch das Bereitstellen der RealTime Suite als Funktionsbibliothek ein Bruch zwischen „den verschiedenen Welten“ vermieden wird.

Das in den meisten Fällen angestrebte Ziel besteht letztlich sicher darin, externe Hardware von der Echtzeitumgebung aus auf verschiedene Weise anzusprechen. Hierzu stehen zunächst sämtliche üblichen Basismechanismen zur Verfügung, um I/O-Register und physischen Speicher anzusprechen und Interrupt-Anforderungen – von PCI- oder PCI-Express-Karten oder auch ISA-Baugruppen – zu bearbeiten. Weiterhin existieren Funktionen für die einfache Benutzung von Kommunikations-Schnittstellen (zum Beispiel seriellen COM-Schnittstellen oder USB-Devices). Des Weiteren werden CAN-Schnittstellenkarten verschiedener Hersteller mit einem allgemeingültigen API mit Echtzeit-Treibern unterstützt. Verschiedene Feldbusse (wie zum Beispiel Profibus) werden mit einer eige-

nen DP-Master-Implementierung direkt unterstützt. Da künftige Automatisierungstechnologien immer häufiger auf Ethernet als Basis aufsetzen, werden echtzeitfähige Netzwerktreiber für die weit verbreiteten Controller von Intel und RealTek bis zu Gigabit-Raten bereitgestellt.

Auf der Basis dieser speziell entwickelten, auf hohe Effizienz und niedrige Latenzzeiten ausgelegten Treiber kann Ethernet auch in Echtzeit-Anwendungen als Kommunikationsstandard dienen. Dies ist nicht nur für anwendungsspezifische Protokolle verwendbar, sondern ermöglicht beispielsweise auch Echtzeit-TCP und -UDP.

Ganz allgemein zeigt sich Industrial Ethernet als der große Trend: Nahezu alle bedeutenden herkömmlichen Feldbusse liegen inzwischen in einer an Ethernet als Transportschicht angepassten Variante auch als Industrial Ethernet vor. Zu den bekanntesten gehören das von Profibus abgeleitete Profinet oder das an CANopen angelehnte EtherCAT.

Herstellen einer PC-basierenden Lösung am Beispiel

EtherCAT unterstützt einen Prozessdatenaustausch mit sehr geringen Zykluszeiten. Der Service-datenaustausch ist ähnlich wie von CANopen her bekannt. In einer EtherCAT-Topologie findet man generell die folgende Gliederung vor, nach der ein Master – das ist die PC-basierte Managementsoftware – die angeschlossenen Slaves verwaltet. Diese

wiederum sind unterteilt in einzelne Objekte, die wiederum mehrere Variablen enthalten können. Diese sind durch einen Namen eindeutig identifiziert und besitzen einen festgelegten Datentyp. Die Werte der Variablen sind les- und/oder beschreibbar. Des Weiteren unterstützt der EtherCAT Master der RealTime Suite auch einige besondere Betriebsarten, wie zum Beispiel Distributed Clocks für zeitlich hochgenaue und synchronisierte Vorgänge. Das System ist zudem Hot-plug-fähig, d. h. die Topologie kann sich während der Laufzeit ändern. Ein Kabelredundanz-Modul erhöht die System-Verfügbarkeit. Alle bisher genannten Funktionsmodule der Echtzeiterweiterung stellen jeweils die spezifische Funktionalität zur Verfügung. Aus Sicht des Programmierers kann es jedoch darüberhinaus wünschenswert sein, die Applikation auf einer noch abstrakteren Ebene zu erstellen. Dies ist um so interessanter, wenn in einer Automatisierungslösung verschiedene Kommunikationsprotokolle, zum Beispiel Profibus und EtherCAT, miteinander verknüpft werden sollen und der Steuerungsablauf dennoch neutral gehalten werden soll. Hierzu wird in einem weiteren Modul eine Soft-SPS nach IEC 61131-3 bereitgestellt. Dadurch ist ein direkter und einfacher Umgang mit Variablen möglich, unabhängig davon, in welcher konkreten Topologie sich diese befinden. Es werden die bekannten PLC-Programmiersprachen Anweisungsliste und Strukturierter Text unterstützt.

U. Jesgarz

Anzeige



B.E.G. LUXOMAT®

FÜR DIE ZUKUNFT UNSERER UMWELT

- Energie sparen mit **B.E.G. Bewegungs- und Präsenzmeldern**
- Für mehr Energieeffizienz in Gebäuden
- Präsenz- und tageslichtorientiert Licht schalten
- Kosten sparen und Umwelt schonen

B.E.G. Brück Electronic GmbH · Schlosserstraße 30 · 51789 Lindlar
Tel.: 02266-90 12 10 · info@beg.de · www.beg-luxomat.com

ENVIRONMENT