

Wie das Realtime-Ethernet-System »EtherCAT« bewährte Kommunikationsprofile nutzt und mit »verteilten Uhren« arbeitet

# Als Antriebsbus geeignet, weil...

Bei der Entstehung des Realtime-Ethernet-Systems »EtherCAT« stand die kombinierte Einsatzfähigkeit für Antriebstechnik und schnelle I/O-Signale im Vordergrund. Im folgenden Fachbeitrag beschreibt ein Vertreter des Entwickler-teams die entsprechenden Eigenschaften des »EtherCAT-Slave-Controllers« und geht im Detail auf die für die Antriebstechnik relevanten Protokoll- und Profimplementierungen ein.

Von Dr. Dirk Janssen

Die speziellen Anforderungen der Antriebstechnik gelten für ein Feldbussystem immer noch als besonders schwierig zu erfüllen. Kurze Zykluszeiten und hohe Synchronität, wie sie für über den Bus geschlossene Regelkreise benötigt werden, lassen sich in bisherigen Systemen nur mit speziellen Antriebsbussen, wie zum Beispiel »SER-



COS interface«, realisieren. Diese Antriebsbusse eignen sich aber nur bedingt für vermeintlich triviale Aufgaben, wie etwa nebenbei auch noch normale I/O-Signale im Zyklus der SPS auszutauschen. Die Folge sind getrennte Feldbusse für Antriebe und I/O-Signale. Soll die Steuerung auch vernetzt werden, kommt noch ein Ethernet-Anschluss hinzu. Für den Anwender ergeben sich dann drei inkompatible Netzwerke, die unterschiedlich verkabelt und parametrierbar sind – wo er doch verständlicherweise nur ein System haben möchte.

## Spezielle Anforderungen aus der Antriebstechnik

Die vermeintlich harten Anforderungen der Antriebstechnik an ein Feldbussystem lassen sich relativ schnell aufzählen: Zyklus-

zeit, Synchronität und Gleichzeitigkeit. Typische Werte für benötigte Zykluszeiten liegen zwischen moderaten 4 ms (zyklische Lagevorgabe mit Lageregelung im Antrieb) und in extremen Fällen 62,5  $\mu$ s (über den Bus geschlossener Stromregelkreis).

Als ausreichende Anforderung an die Synchronität wird in der Antriebstechnik häufig eine Mikrosekunde angegeben. Da der Begriff der Synchronität in diesem Zusammenhang nicht klar definiert ist, sollte zusätzlich die Gleichzeitigkeit bewertet werden. Während die Synchronität den zeitlichen Jitter der Abarbeitung der Funktionen in den beteiligten Teilnehmern (Antriebe und Steuerung) angibt, definiert die Gleichzeitigkeit das Maß des zeitlichen Versatzes dieser Funktionen. Die Synchronität ist für den einzelnen Teilnehmer wichtig, damit eigene, unterlagerte Regelkreise sich auf das zyklische Signal entsprechend genau synchronisieren können. Die Gleichzeitigkeit erlaubt zudem, verteilte Teilnehmer an einer gemeinsamen Aufgabe, mit der absolut selben Zeitbasis, arbeiten zu lassen. Im Vergleich zu klassischer SPS-Technik, mit Zykluszeiten von 20–30 ms mit freilaufender Prozessdatenkommunikation, sind die beschriebenen Anforderungen schon deutlich höher. Moderne SPS-Systeme (z. B. PC-basierte Soft-SPS) erreichen inzwischen aber mindestens die gleichen Zykluszeiten und tauschen ihre I/O-Signale ebenfalls synchron aus. Da eine SPS neben der eigentlichen Steuerungstechnik immer mehr Funktionen aus den Bereichen Überwachung und Messtechnik übernimmt, sind hier zukünftig sogar noch deutlich höhere Anforderungen zu erwarten. Maschinen- und Werkzeugsicherungssysteme, die heute mit spezieller Hardware realisiert werden, lassen sich so einfach in die SPS als Softwarelösung integrieren. Während in der Antriebstechnik die benötigten Zykluszeiten und Synchronitäten durch die Trägheit der anzusteuern Mechaniken begrenzt sind, werden zukünftig die Anforderungen aus der Messtechnik das Einsatzspektrum eines Feldbussystems bestimmen.

## Verteilte Uhren

Synchronität und Gleichzeitigkeit zwischen verteilten Systemen lassen sich auf verschiedene Art – zumindest ansatzweise – herstellen. Ein häufig angewendetes Verfahren beruht auf dem zyklischen Versenden eines Synchronisationssignals, das von einem Teilnehmer – meist dem Master – ausgesendet und quasi gleichzeitig von allen anderen empfangen wird. Das Verfahren setzt voraus, dass das Bussystem immer frei ist, wenn das Synchronisationssignal versendet werden soll, und geht von einem verzögerungsfreien Empfang in den Teilnehmern aus. Aufgrund von Laufzeiten in den Teilnehmern und in den Infrastrukturkomponenten (Kabel, Switches, Netzwerk-Controller etc.) ist die Gleichzeitigkeit aber nicht gegeben. Ein 100 m langes Kabel erzeugt bereits eine Verzögerung von ca. einer halben Mikrosekunde. Das Weiterleiten des Signals durch einen Switch kann mehrere Mikrosekunden dauern. Wird das Bussystem für mehrere parallele Kommunikationen benutzt (für Antriebssteuerung, I/O-Signale und übergeordnete Leitsysteme) ist selbst das exakte Versenden des Synchronisationssignals nur schwer sicherzustellen. EtherCAT nutzt daher einen anderen Ansatz, der auf so genannten »verteilten Uhren« basiert: Alle Teilnehmer besitzen eine eigenständige Uhr, auf Basis derer die lokalen Zyklen und Ereignisse ablaufen. Entscheidend dabei ist, dass alle Uhren gleich schnell laufen und die gleiche Basiszeit besitzen. Eine im EtherCAT-Slave-Controller (ESC) integrierte Regelung stellt sicher, dass alle Uhren sich an einer Referenz-Uhr orientieren und unabhängig von Temperatur und Herstell-Toleranzen synchron laufen.

Um zusätzlich die Gleichzeitigkeit herzustellen, müssen alle Uhren die gleiche Basiszeit erhalten. EtherCAT definiert hierzu eine Systemzeit mit einer Auflösung von 1 ns. Die Systemzeit wird als 64-Bit-Integer direkt vom ESC verwaltet und stellt jederzeit den absoluten zeitlichen Bezug (bezogen auf

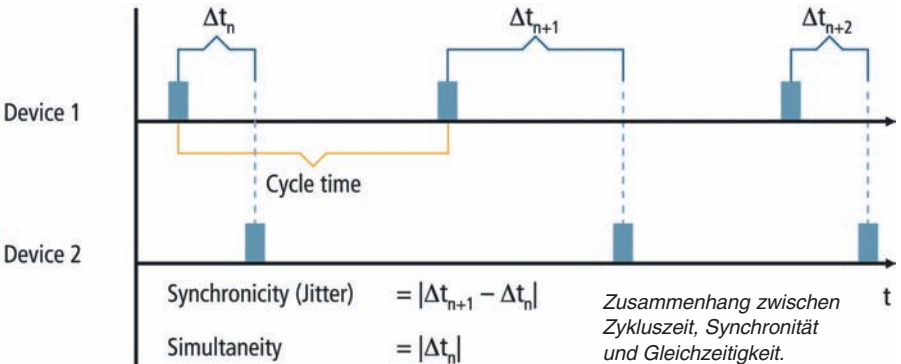
den 1. 1. 2000 um 0:00 Uhr) zu Ereignissen im gesamten Netzwerk her. Da durch die Regelung alle verteilten Uhren synchronisiert sind, muss die Systemzeit nur einmalig initialisiert werden.

Um die Systemzeit entsprechend genau in allen Teilnehmern initialisieren zu können, wird eine vom ESC unterstützte Laufzeitmessung durchgeführt, die die Signallaufzeiten zwischen allen EtherCAT-Teilnehmern ermittelt und die dann bei der Verteilung der Systemzeit berücksichtigt werden. Dazu ist unter anderen die genaue Kenntnis der installierten Topologie notwendig, die bei EtherCAT sehr flexibel (Linie, Stern, Baum) aufgebaut sein kann. Die Topologie ist aber vollständig auslesbar – ebenfalls vom ESC unterstützt. Eine statistische Mittelwertbildung hilft dabei, den im Kommunikationssystem vorhandenen Jitter zu kompensieren. Für die Nutzung der verteilten Uhren und deren Initialisierung ist keine Unterstützung durch den Anwendungsprozessor im Teilnehmer notwendig, da dies vollständig in den EtherCAT-Slave-Controllern durchgeführt wird.

Basierend auf den verteilten Uhren bietet

Systemzeiten der einzelnen Teilnehmer initialisiert werden können. Die bereits angesprochene Mikrosekunde für die Synchronität wird bereits annähernd von klassischen – auf Antriebstechnik spezialisierten – Feldbussen erreicht. Die erreichte Gleichzeitigkeit liegt aber, abhängig von der Anzahl der Teilnehmer, häufig deutlich darüber. Selbst das Kommunikationssystem SERCOS interface vernachlässigt die erforderliche Gleichzeitigkeit und geht davon aus, dass das entsprechende Synchronisationssignal in allen Teilnehmern zur selben Zeit empfangen wird. In der Realität wird das Signal aber von einem Teilnehmer zum nächsten weitergeleitet und um ca. 1  $\mu$ s verzögert. Selbst bei nur 20 Teilnehmern ergibt sich daher bereits ein entsprechender Fehler von ca. 20  $\mu$ s. Dieser Fehler kann auch nicht automatisch kompensiert werden, da erstens die Topologie nicht auslesbar ist, zweitens die Laufzeit nicht messbar ist und drittens keine Vorkehrungen im entsprechenden Protokoll getroffen wurden.

EtherCAT ist ausgelegt, um nicht nur 20 Teilnehmer, sondern mehrere hundert bzw.



der ESC eine Reihe von Funktionen und Signalen, die die hochgenaue Zeitbasis für die jeweilige Anwendung (z. B. eines Antriebsreglers) zur Verfügung stellt. Gearbeitet wird dabei immer mit der absoluten Systemzeit, die automatisch den zeitlichen Bezug verteilter Teilnehmer sicherstellt. Durch Synchronisationssignale, die beispielsweise mit Capture/Compare-Einheiten des Anwendungsprozessors verbunden sind, wird die hochgenaue Zeitbasis in der Anwendung verfügbar. Der ESC stellt auch Synchronisationseingänge zur Verfügung, mit deren Hilfe die Systemzeit für externe Ereignisse gelatcht wird. Durch Verwendung der absoluten Systemzeit können auch weit verteilte Ereignisse gemeinsam bewertet werden.

Entscheidend für die Einsetzbarkeit der verteilten Uhren sind die Qualität der Regelung und die Genauigkeit, mit der die

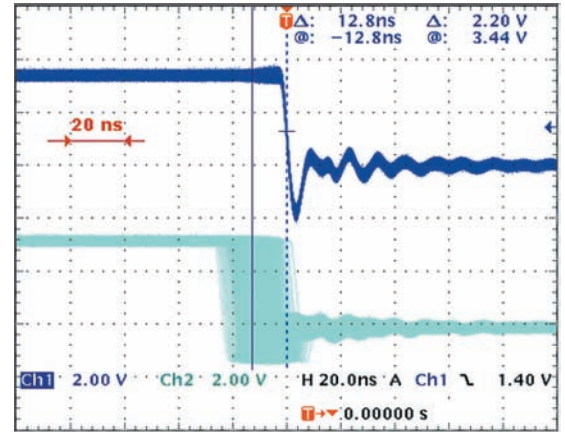
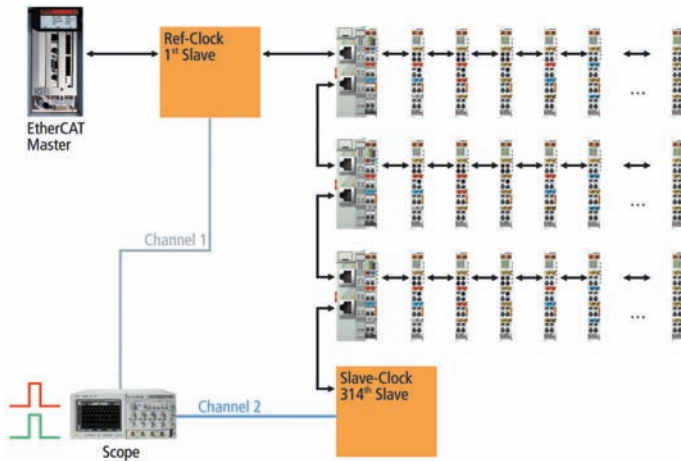
tausend Teilnehmer, zu unterstützen. Die bereits aufgeführten Mechanismen erlauben trotzdem Ergebnisse, die um mehrere Größenordnungen besser sind.

## Multiprotokollfähigkeit

Weitere wichtige Kriterien eines Feldbusystems zur Unterstützung der Antriebstechnik sind das verwendete Kommunikationsprotokoll und -profil, die für die Kompatibilität und den effizienten Datenaustausch zwischen Steuerung und Antrieb verantwortlich sind. Statt hier das Rad neu zu erfinden, setzt EtherCAT in diesem Bereich auf bewährte Technik.

Die Kommunikationsanforderungen moderner Feldbusse (Prozessdaten, Parameterdaten, paralleles TCP/IP, Firmwareupdates, Routing zu unterlagerten Bussystemen, etc.) werden von keinem verfü-





Testaufbau für eine Langzeit-Scope-Aufnahme: Zwischen den beiden EtherCAT-Teilnehmer befinden sich im Testaufbau über 300 weitere Teilnehmer und insgesamt ca. 120 m Kabel, die eine realistische Anwendung repräsentieren. Das Scope triggert auf das Signal des ersten Teilnehmers und zeigt unten das Signal des zweiten. Die Breite der Signalfanke (Nachleuchtzeit auf unendlich) entspricht dem Synchronisationsfehler und liegt bei ca.  $\pm 20$  ns. Die Abweichung des Mittelwertes vom Triggerpunkt des ersten Teilnehmers entspricht dem Gleichzeitigkeitsfehler und liegt hier bei ca. 12 ns. Umfangreiche Messungen haben ergeben, dass sowohl der Synchronisationsfehler als auch der Gleichzeitigkeitsfehler – selbst bei großen Netzwerken – deutlich unter 100 ns liegen.

baren Protokoll alleine unterstützt. Daher setzt EtherCAT auf Multiprotokollfähigkeit und führt die unterschiedlichen Protokolle in einer einheitlichen Mailbox zusammen. Dies erleichtert, bestehende Geräte schnell und vollständig auf EtherCAT umzusetzen. Für die Antriebstechnik relevant sind die Protokolle »CANopen over EtherCAT« (CoE) und »SERCOS Device Profile over EtherCAT« (SoE), die es jeweils ermöglichen, die Vorteile von EtherCAT bezüglich der Übertragungseigenschaften mit den bewährten profilspezifischen Antriebsfunktionen zu kombinieren. Die Protokolle »Ethernet over EtherCAT« (EoE) und »File Access over EtherCAT« (FoE) ermöglichen optional zum Beispiel einen Webserver im Antrieb zu integrieren oder die Firmware bzw. Kurvenscheibentabellen effizient über den Bus auszutauschen. Mit dem spezifi-

zierten CANopen-over-EtherCAT (CoE)-Protokoll wird die Nutzung der gesamten CANopen-Profilfamilie über EtherCAT ermöglicht. Das SDO-Protokoll ist direkt übernommen, sodass bestehende CANopen-Stacks quasi ohne Änderung verwendet werden können. Optional sind Erweiterungen definiert, die einerseits die 8-Byte-Beschränkung aufheben und andererseits die vollständige Auslesbarkeit des Objektverzeichnisses ermöglichen. Die Prozessdaten werden in Prozessdatenobjekten (PDO) organisiert, die mit den effizienten Mitteln von EtherCAT übertragen werden – eine 8-Byte-Beschränkung besteht natürlich nicht mehr. Alle CANopen-Profil – und damit auch das Antriebsprofil (DS 402) – sind vollständig nutzbar und darauf basierende Geräte können sehr einfach auf EtherCAT übertragen werden.

Die EtherCAT-Slave-State-Machine entspricht – bis auf wenige Details – der CANopen-State-Machine, sodass auch hier die notwendigen Änderungen überschaubar bleiben. Um ein eindeutigeres Anlaufverhalten zu ermöglichen, ist ein weiterer State namens »Safe-Operational« definiert, in den bereits gültige Eingänge übertragen werden, während die Ausgänge noch im sicheren Zustand verbleiben. Das Protokoll SERCOS Device Profile over EtherCAT (SoE) erlaubt die Nutzung des sehr bewährten, auf die anspruchsvolle Antriebstechnik spezialisierten und in der IEC 61491 genormten SERCOS-Geräte-Profiles. Der SERCOS-Servicekanal, und damit der Zugriff auf alle antriebsinternen Parameter und Funktionen, wird auf die EtherCAT-Mailbox abgebildet. Auch hier stehen sowohl die Kompatibilität zum bestehen-

den Protokoll (Zugriff auf Wert, Attribute, Namen, Einheiten etc. der SERCOS-Identifizier) als auch die Erweiterungsmöglichkeit bezüglich der Datenlängenbeschränkung im Vordergrund. Die Prozessdaten, bei SERCOS in Form von AT- und MDT-Daten, werden wiederum mit den Mitteln des EtherCAT-Slave-Controllers übertragen; das entsprechende Mapping erfolgt SERCOS-konform über die Identifier S-0-0015, S-0-0016 und S-0-0024.

Zur Synchronisierung werden – wie auch beim CoE-Protokoll – die bereits erläuterten Synchronisationseigenschaften des EtherCAT-Slave-Controllers genutzt. Diese

gleichbar. Es findet die notwendige Synchronisierung statt, allerdings müssen bei EtherCAT bereits gültige Eingänge übertragen werden. »Operational« entspricht wiederum exakt der Phase 4, in der der normale, zyklische Datenaustausch stattfindet.

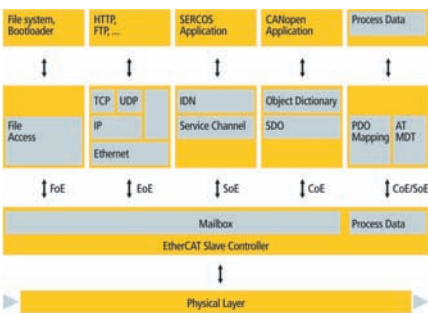
Die EtherCAT-Slave-State-Machine ist, wie der Name schon sagt, auf einen Slave bezogen und erlaubt somit – im Gegensatz zu SERCOS – einzelne Antriebe unabhängig von anderen Teilnehmern neu zu parametrisieren und in Betrieb zu nehmen.

## Fazit

EtherCAT ist kein reiner Antriebsbus, aber erfüllt die entsprechenden Anforderungen sehr gut. Eine Trennung von Antriebs-, I/O- und Kommunikationsbus ist nicht mehr notwendig. Selbst anspruchsvollere Aufgaben, wie etwa in der Messtechnik, lassen sich integrieren und erlauben, neue Funktionalitäten mit »klassischer« Steuerungstechnik zu nutzen. Durch Verwendung bewährter Kommunikationsprofile ist eine Migration bestehender Geräte und Applikationen sehr einfach möglich. Insbesondere in der Antriebstechnik haben sich hier einige wenige Profile über Jahre hinweg entwickelt und bewährt. Außerdem können so die gesamte Toolkette und vorhandene Erfahrungen zur Parametrierung entsprechender Antriebe erhalten bleiben.

■ [www.ethercat.org](http://www.ethercat.org)

**Zum Autor:** Dr. Dirk Janssen ist Leiter der Grundlagensoftware bei der deutschen Firmenzentrale von Beckhoff in Verl.



Multiprotokollfähigkeit der EtherCAT-Mailbox.

umfassen – entsprechend qualitativ verbessert – auch die von Standard-SERCOS, sodass hier eine Umsetzung entsprechend leicht fällt.

Die bereits erläuterte EtherCAT-Slave-State-Machine lässt sich ebenfalls gut auf die Phasen des SERCOS-Protokolls abbilden. »Pre-Operational« entspricht SERCOS-Phase 2 und erlaubt Servicekanalkommunikation ohne Prozessdatenaustausch. »Safe-Operation« ist mit Phase 3 ver-