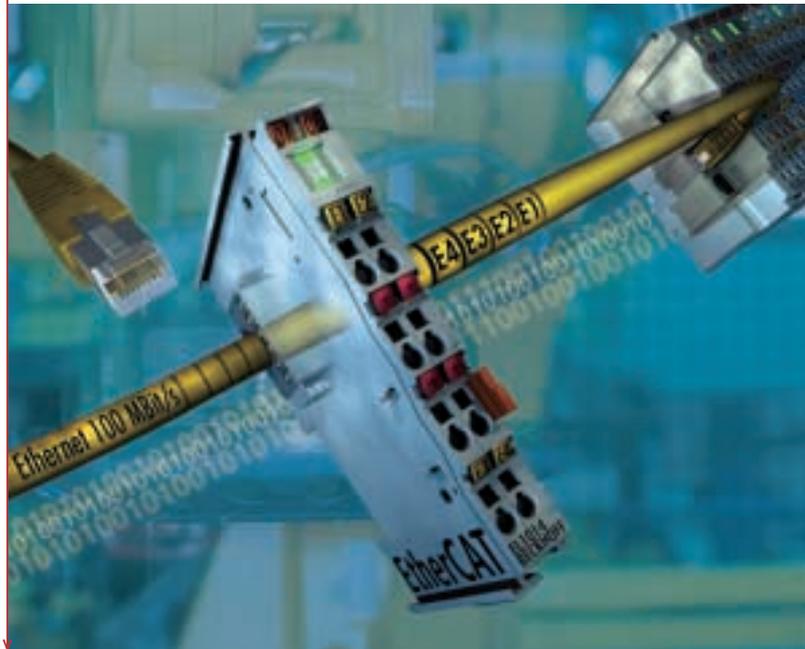


Die technische Antwort auf die Frage: „Warum EtherCAT?“

EtherCAT – Der Ethernet Feldbus



→ EtherCAT – die Ethernet-Echtzeit-Lösung von Beckhoff wurde erstmalig auf der Hannover Messe 2003 vorgestellt. Das Konzept, die Systemeigenschaften und die damit verbundenen Vorteile für zukünftige Automatisierungslösungen wurden bereits in der letzten PC-Control Ausgabe 02/03 beschrieben. In dem folgenden Beitrag erläutern Dirk Janssen und Holger Büttner, zwei Beckhoff-Mitarbeiter aus dem EtherCAT-Entwicklungsteam, detailliert das Funktionsprinzip und die technischen Eigenschaften von EtherCAT.

In der weltweiten Automatisierungstechnik nimmt der Einsatz von Ethernet als physikalische Kommunikationsschicht rasant zu. Beginnend mit der Leitebene, über die Fabrikvernetzung bis hin zur Intersteuerungskommunikation ist Ethernet bereits etabliert und aufgrund der, aus der Office-Welt bekannten, hohen Komponenten-Stückzahlen vergleichsweise leistungsfähig und kostengünstig.

In der Feldebene dominieren weiterhin die in den 90er Jahren entwickelten Feldbusse. Mit unterschiedlichen Schwerpunkten haben sich die Feldbusse ihr jeweiliges Marktsegment geschaffen und mit maximalen Bruttoübertragungsraten von 0,5–16 MBaud sind sie auch weiterhin für viele Anwendungen ausreichend. Die, im Vergleich zur allgemeinen IT-Welt, verschwindend geringen Stückzahlen führen jedoch zu hohen Gesamtkosten – sowohl in der Steuerung als auch in den Feldgeräten und insbesondere in der Verkabelung. Die notwendige Flexibilität bezüglich der Signalgranularität wurde und wird durch modulare Feldgeräte erreicht, die einen unterlagerten Sub-Bus betreiben. Ein Sub-Bus erhöht die Komplexität des Gesamtsystems und hat signifikante Performance-Nachteile bezüglich Totzeiten und erreichbarer Zykluszeit.

Warum EtherCAT?

Während das Moore'sche Gesetz zumindest bei den PC-basierten Steuerungen in den letzten Jahren weiterhin Gültigkeit behalten hat und in Zukunft behalten wird (Verdoppelung der Leistungsfähigkeit ca. alle 2 Jahre), gab es bei den Feldbussen keine nennenswerten Weiterentwicklungen. Stattdessen wurde Ethernet mehrheitlich als kommende Ergänzung bzw. Ersatz für die „proprietäre“ Feldbustechnik sowohl von Feldbus-Organisationen als auch von „großen“ Automatisierungsfirmen ausgewählt, und entsprechende Standards wurden entworfen.

An dieser Stelle muss sich eine Firma wie Beckhoff fragen lassen: „Wieso noch ein weiterer Ethernet-Ansatz?“ Die kurze, aber prägnante Antwort lautet: „EtherCAT geht andere Wege und ist aktuell mit Abstand der leistungsfähigste und der am besten auf automatisierungstechnische Anforderungen zugeschnittene Ethernet-Ansatz!“ Eine fundierte, technisch begründete Antwort darauf folgt in diesem Artikel.

Ethernet macht auch Probleme

Ethernet hat im Vergleich zu den existierenden Feldbussen nicht nur Vorteile. Die für den Einsatz in der Automatisierungstechnik eher ungünstigen Eigenschaften von Ethernet müssen daher sehr genau betrachtet und möglichst optimal umgangen werden. Darauf beruhen auch die wesentlichen Unterschiede der verschiedenen Ansätze, Ethernet für die Automatisierungstechnik anzupassen:

- | Hoher Overhead bei der Kommunikation mit Teilnehmern, die zwar häufig aber nur wenige Daten austauschen müssen.
- | Hohe Anschaltkosten pro Teilnehmer (Trafo, Phy, Mac und notwendige Prozessor-Performance) verglichen mit klassischen Feldbusanschlüssen.
- | Mangelnde Echtzeitfähigkeit, die bei genauerer Betrachtung aber weniger durch Ethernet als Übertragungsmedium verursacht wird sondern auf die ungünstigen Laufzeiten in den Software-Stacks zurückgeht.
- | Ungünstige Topologie: Die inzwischen bei Ethernet übliche sternförmige Topologie ist bei der Anlagenverdrahtung eher ungünstig und führt entweder zu hohem Verkabelungsaufwand oder zu stark kaskadierten Kommunikationsbeziehungen.

Ethernet ist auch nicht gleich Ethernet. Neben den unterschiedlichen Übertragungsgeschwindigkeiten von 10, 100 und 1000 MBaud muss auch zwischen Halb- und Vollduplex-Kommunikation unterschieden werden. Halbduplex – also die Datenübertragung, die abwechselnd in die eine und in die andere Richtung geht – überträgt, im Vergleich zu Vollduplex, weniger als die Hälfte, da nach der Übertragung in die eine Richtung Pausenzeiten eingehalten werden müssen, um die Lauf- und Reaktionszeiten zu kompensieren. Bei Halbduplex-Übertragungen können Kollisionen auftreten, die zwar auf den untersten Ebenen abgefangen und durch Wiederholungen beseitigt werden, für eine deterministische Übertragung aber nicht akzeptabel sind. Daher müssen die Kollisionen auf höheren Ebenen vermieden werden, was aber wiederum nicht ohne weitere Verschlechterung der Nutzdatenrate möglich ist.

Ziele bei der Entwicklung von EtherCAT

PC-basierte Steuerungstechnik kann aufgrund der Leistungsfähigkeit des PC-Systems eine eher zentrale oder – genauer ausgedrückt – eine hierarchische Steuerungsphilosophie unterstützen, bei der alle auf einer Ebene relevanten Informationen gleichzeitig in einer Steuerung bekannt sind und miteinander kombiniert werden können. Dies erleichtert nicht nur die Konfiguration des Systems, sondern ermöglicht auch den Einsatz intelligenterer Steuerungsalgorithmen. Das Kommunikationssystem ist dabei „nur noch“ für den schnellen Transport der Prozessdaten in und aus der Steuerung verantwortlich. Ein Master-Slave-Kommunikationssystem ist für diesen Zweck die beste Wahl. Bei der Entwicklung von EtherCAT standen die folgenden Hauptziele im Vordergrund:

- | Breite Einsatzbarkeit. Als EtherCAT-Master soll jede Steuerung mit einem handelsüblichen Ethernet-Controller eingesetzt werden können. Angefangen von einem kleinen 16 Bit μ C bis hin zu PC-Systemen mit 3 GHz Prozessoren soll jeder Rechner ohne spezielle Anschaltung zur EtherCAT-Steuerung werden können.
- | Vollständige Konformität zum Ethernet-Standard. EtherCAT soll mit anderen Ethernet-Geräten und -Protokollen am selben Bus koexistieren. Standard-Struktur-Komponenten wie Ethernet-Switches sollen für EtherCAT einsetzbar sein.
- | Kleinstmögliche Teilnehmer-Granularität ohne unterlagerten Sub-Bus. Als EtherCAT-Slave sollen sowohl komplexere Knoten als auch 2-Bit-I/Os wirtschaftlich eingesetzt werden können.
- | Höchstmögliche Effizienz. Die Bandbreite von Ethernet soll möglichst vollständig für Nutzdatentransfers zur Verfügung stehen.
- | Kurze Zykluszeiten. Mögliche Zykluszeiten deutlich unter 100 μ s sollen neue Anwendungsgebiete erschließen, wie z.B. das Schließen unterster Regelkreise in der Antriebstechnik.
- | Höchste Deterministik, auch ohne die Basis absoluter Telegramm-Sendegenauigkeit.

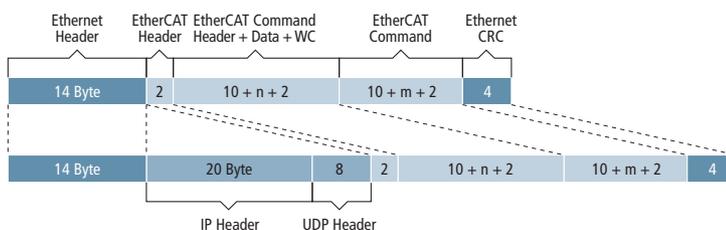
EtherCAT-Funktionsprinzip

Aus Ethernet-Sicht ist ein EtherCAT-Bus nichts anderes als ein einzelner großer Ethernet-Teilnehmer. Dieser „Teilnehmer“ empfängt und sendet Ethernet-Telegramme. Innerhalb des „Teilnehmers“ befindet sich aber kein Ethernet-Controller mit nachgeschaltetem μ -Prozessor, sondern eine Vielzahl von EtherCAT-Slaves. Diese verarbeiten die einlaufenden Telegramme im Durchfluss und nehmen die für sie bestimmten Nutzdaten heraus bzw. blenden sie ein und leiten das Telegramm an den nächsten EtherCAT-Slave weiter. Der letzte EtherCAT-Slave schickt das bereits vollständig verarbeitete Telegramm zurück, so dass es vom ersten Slave – quasi als Antwort-Telegramm – zur Steuerung zurückgeschickt wird. Es wird dabei ausgenutzt, dass Ethernet eine getrennte Übertragung in Hin- und Rück-Richtung (Tx- und Rx-Leitungen) besitzt und im Vollduplex-Modus arbeitet. Natürlich kann, wie bei jedem anderen Ethernet-Teilnehmer auch, eine direkte Kommunikation ohne Switch mit einem „gedrehten“ Ethernet-Kabel aufgebaut werden, wodurch ein reines EtherCAT-System entsteht.

Telegrammverarbeitung

Die Verarbeitung der Telegramme findet im Durchlauf statt. Während die Telegramme, nur um wenige Bits verzögert, bereits weitergeschickt werden, erkennt

der Slave für sich bestimmte Kommandos und führt sie entsprechend aus. Die Verarbeitung findet in Hardware statt und ist daher unabhängig von den Reaktionszeiten eventuell angeschlossener μ -Prozessoren. Jeder Teilnehmer besitzt dabei einen adressierbaren Speicherbereich von 64 kByte innerhalb dessen gelesen, geschrieben oder gleichzeitig geschrieben und gelesen werden kann. Innerhalb eines Ethernet-Telegramms können mehrere EtherCAT-Kommandos eingebettet werden, die jeweils individuelle Teilnehmer und/oder Speicherbereiche ansprechen. Die EtherCAT-Kommandos werden im Datenbereich eines Ethernet-Telegramms transportiert und können entweder über einen speziellen Ether-Type oder per UDP/IP kodiert sein.



EtherCAT-Telegrammaufbau (ohne und mit UDP)

Die erste Variante mit speziellem Ether-Type ist zwar auf ein Ethernet-Subnetz begrenzt, d. h. entsprechende Telegramme werden von Routern nicht weitergeleitet, für Steuerungsaufgaben stellt dies in der Regel jedoch keine Einschränkung dar. Als Adressierung wird die Ethernet-MAC-Adresse des ersten Teilnehmers genutzt. Hierbei wird ein spezieller erster EtherCAT-Teilnehmer benötigt, der im Falle einer direkten Ansteuerung ohne Switch entfallen kann.

Die zweite Variante per UDP/IP erzeugt einen geringfügig größeren Overhead (IP- und UDP-Header), ermöglicht aber, bei weniger zeitkritischen Anwendungen, einerseits das normale IP-Routing zu nutzen und andererseits die häufig bereits vorhandenen TCP/IP-Stacks auf der Masterseite zu verwenden. Jedes EtherCAT-Kommando besteht aus einem EtherCAT-Header, dem Datenbereich und einem anschließenden Zählerfeld (Working-Counter), das von allen EtherCAT-Teilnehmern inkrementiert wird, die durch das EtherCAT-Kommando angesprochen wurden und entsprechende Daten ausgetauscht haben.

Fieldbus-Memory-Management-Unit (FMMU)

Mit Hilfe des beschriebenen Telegrammaufbaus können bereits mehrere EtherCAT-Teilnehmer durch ein einzelnes Ethernet-Telegramm mit mehreren EtherCAT-Kommandos angesprochen werden, was zu einer deutlichen Verbesserung der Nutzdatenrate führt. Für den Fall einer 2-Bit-Eingangsklemme, die eben genau 2 Bit Nutzdaten besitzt, ist der Overhead eines einzelnen EtherCAT-Kommandos aber weiterhin deutlich zu groß.

Die Fieldbus-Memory-Management-Unit beseitigt das Problem und ermöglicht eine nahezu 100 prozentige Nutzdatenrate – auch bei Teilnehmern, die, wie beschrieben, nur 2 Bit Nutzdaten aufweisen. Ähnlich wie eine Memory-Management-Unit (MMU) in modernen Prozessoren, übersetzt die FMMU eine logische Adresse anhand einer internen Tabelle in eine physikalische. Die FMMU ist im EtherCAT-Slave Asic integriert und ermöglicht für jeden Teilnehmer individuell ein entsprechendes Adressmapping.

Übertragungsschicht	Kabel	Max. Länge	Kosten (Anschaltung/lfd. Meter)	Konfektionier- barkeit	Verzögerung (pro Anschaltung in µs)
Ethernet (100 Base TX)	CAT 5 (Kupfer)	100 m	o / ++ (+ ¹)	+	ca. 1
Ethernet (100 Base FX)	LWL (Glasfaser)	100 km	- / --	-	ca. 1
E-Bus (Kupfer)	CAT 5 (Kupfer)	10 m	++ / ++ (+ ¹)	+	0
E-Bus (LWL, in Planung)	LWL (Plastikfaser)	100 m	+ / +	++	0
Eigenschaften der verschiedenen physikalischen Schichten (¹ industrielle Kabelführung d.h. schleppkettentauglich, ölfest)					

Im Unterschied zu prozessorinternen MMUs, die komplette Speicherseiten (im Bereich von 4 kByte) mappen, unterstützt die FMMU auch bitweises mappen. Dadurch können die zwei Bit der Eingangsklemme individuell in einen beliebigen Bereich eines logischen Adressraumes eingeblendet werden. Wird nun ein EtherCAT-Kommando verschickt das, anstatt einen speziellen EtherCAT-Teilnehmer anzusprechen, einen bestimmten logischen Speicherbereich liest oder schreibt, blendet die 2-Bit-Eingangsklemme ihre Daten an der richtigen Stelle in den Datenbereich ein. Alle anderen Klemmen, die ebenfalls einen Adressmatch in ihrer eigenen FMMU feststellen, blenden ebenfalls ihre Daten ein, so dass mit einem einzelnen EtherCAT-Kommando viele Teilnehmer gleichzeitig angesprochen werden.

Da die FMMU in jedem Teilnehmer vorhanden ist und individuell konfiguriert wird, kann der Master in der Initialisierungsphase bereits komplette Prozessabbilder zusammenstellen und anschließend mit einem einzelnen EtherCAT-Kommando austauschen. Es wird kein zusätzliches Mapping mehr im Master benötigt, so dass die Prozessdaten direkt den unterschiedlichen Steuerungstasks (SPS, NC etc.) zugeordnet werden können. Jede Task kann ihr eigenes Prozessabbild erstellen und in ihrem eigenen zeitlichen Rahmen austauschen. Die physikalische Reihenfolge der EtherCAT-Teilnehmer ist dabei völlig unabhängig und spielt nur in der allerersten Initialisierungsphase eine Rolle.

Der logische Adressraum für die FMMUs beträgt 4 GByte. Aus Sicht des Masters kann ein EtherCAT System daher als großer verteilter Speicher (Distributed Memory) angesehen werden, in den wahlfrei geschrieben und gelesen werden kann. EtherCAT ist daher eine Art serielle Backplane für Automatisierungssysteme, die für große aber auch ganz kleine Automatisierungsgeräte die Anbindung an verteilte Prozessdaten ermöglicht. Über einen Standard-Ethernet-Controller und Standard-Ethernetkabel (CAT 5) können quasi beliebig viele und beliebig verteilte I/O-Kanäle an Automatisierungsgeräte angeschlossen werden, auf die mit hoher Bandbreite, geringster Verzögerung und nahezu optimaler Nutzdatenrate zugegriffen werden kann.

Physikalische Schicht

Für Ethernet existiert eine Vielzahl unterschiedlicher physikalischer Medien; angefangen vom in die Jahre gekommenem „Yellow Cable“ bis zu Hochgeschwindigkeits-Glasfaser-Strecken. Bei dem von EtherCAT verwendeten 100 Mbaud Ethernet gibt es – neben dem klassischen CAT 5 Kupferkabel (100 Base TX) – auch Lichtwellenleiter-Übertragungsschichten (100 Base FX). Da EtherCAT vollständig kompatibel zu Ethernet ist, können alle entsprechenden Übertragungsschichten

genutzt werden. Bei EtherCAT erfolgen häufig Übertragungen auf sehr kurzer Strecke, z. B. zwischen zwei EtherCAT-Klemmen im selben Klemmenblock, so dass zusätzlich eine weitere Übertragungsschicht entwickelt wurde – der E-Bus. Der E-Bus basiert auf einer LVDS-Übertragung (Low-Voltage-Differential-Signal, IEEE Standard P1596.3-1995) und ist neben der Klemmen-Kommunikation auch für kostengünstige „Fernübertragungen“ bis ca. 10 m geeignet.

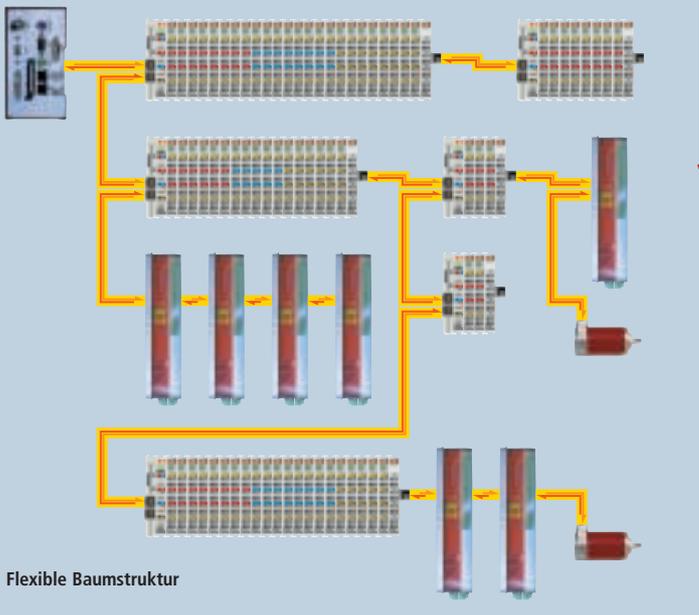
Da es sich bei den übertragenen Daten jederzeit um vollständig kompatible Ethernet-Telegramme handelt, kann die physikalische Übertragungsschicht an jeder Stelle und beliebig oft gewechselt werden. Bezogen auf das Beispiel mit den verschiedenen Schaltschränken und Maschinenmodulen, kann die jeweils kostengünstigste Übertragungsschicht genutzt werden. Innerhalb eines Schaltschranks reicht der E-Bus; zwischen diesem und den Maschinenmodulen kommt die normale Ethernet-Kupfer-Physik mit bis zu 100 m zum Einsatz. Bei noch größeren Entfernungen oder extremen EMV-Belastungen kann wiederum an jeder Stelle auf LWL-Physik umgesetzt werden.

Einzige Voraussetzung an das Übertragungsmedium ist die Vollduplex-Fähigkeit, da EtherCAT so schnell reagiert, dass in der Regel bereits die Antwort zum Master zurückgesendet wird, während der Master noch die letzten Bytes seiner Anfrage absendet. Halbduplex-Übertragungsschichten, wie sie z. B. bei der Funkübertragung genutzt werden, würden dabei Kollisionen erzeugen und können für EtherCAT nicht eingesetzt werden. EtherCAT nutzt in der aktuellen Implementierung 100 Mbaud Ethernet. Das Übertragungsprinzip ist davon aber unabhängig und kann zukünftig ohne Änderung auf 1 Gbaud oder mehr angewendet werden.

Topologie

Die Topologie eines Kommunikationssystems ist mit ausschlaggebend für den erfolgreichen Einsatz in der Automatisierungsindustrie. Die Topologie hat großen Einfluss auf die Verkabelungsaufwendungen, Diagnose-Eigenschaften, Redundanz-Möglichkeiten und Hot-Plug-and-Play-Eigenschaften. Die bei Standard-Ethernet (100 Base TX) übliche Sternverkabelung hat zwar Vorteile bezüglich Hot-Plug-and-Play und Redundanz, die Verkabelungsaufwendungen und notwendigen Switche sind aber in verteilten Anwendungen mit vielen Teilnehmern kaum akzeptabel.

Bei EtherCAT stellen die Slaves logisch gesehen einen offenen Ringbus dar. Am offenen Ende schickt der Master, entweder direkt oder über Standard-Ethernet-Switche, Telegramme hinein und erhält sie am anderen Ende bearbeitet wieder zurück. Alle Telegramme werden vom ersten Teilnehmer an die nächsten weiter-



Flexible Baumstruktur

geleitet und vom letzten wird das Telegramm wieder zurück zum Master gesendet. Da ein normales Ethernet-Kabel bidirektional aufgebaut ist (getrennte Tx und Rx Leitungen) und auch alle EtherCAT-Slaves in der Rückrichtung übertragen können, ergibt sich ein physikalischer Strang.

Durch Abzweigungen, die prinzipiell an jeder Stelle möglich sind, kann aus der Strangstruktur eine flexible Baumstruktur aufgebaut werden. Eine Baumstruktur ermöglicht einfachste Verkabelungen; so können einzelne Äste z.B. in Schaltschränke oder Maschinenmodule verzweigen, während der Hauptstrang von einem Modul zum nächsten läuft.

Telegramm-Übertragung

Jeder EtherCAT-Slave hat zwei Rx- und Tx-Schnittstellen, über die die Telegramme jeweils in der Hin-Richtung und in der Rück-Richtung geleitet werden. Die Auswertung der Telegramme erfolgt immer in Hin-Richtung, die Rück-Richtung dient zum Verstärken und Regenerieren des Signals und zum Lokalisieren und Schließen von Leitungsunterbrechungen (Bild a).

Ein EtherCAT-Slave kann erkennen, ob auf seiner Hin- bzw. Rück-Leitung ein Trägersignal anliegt. Weiterhin besitzt jeder EtherCAT-Slave die Eigenschaft, den Datenfluss so umschalten zu können, dass ein über die Hin-Leitung empfangenes Telegramm sowohl über die Tx-Schnittstelle der Hin-Leitung als auch der Rück-Leitung gesendet werden kann. Und umgekehrt kann ein über die Rück-Leitung empfangenes Telegramm über die Tx-Schnittstelle der Rück- oder der Hin-Leitung geschickt werden (Bild b).

Wenn der EtherCAT-Slave auf seiner Rückleitung kein Trägersignal mehr erkennt,

schaltet er die Rx-Schnittstelle der Hin-Leitung auf die Tx-Schnittstelle der Rück-Leitung kurz, d. h. ein über die Hin-Leitung empfangenes Telegramm wird bearbeitet und dann über die Tx-Schnittstelle der Rückleitung zurückgesendet. Durch diese Funktionalität werden keine zusätzlichen Abschlussmodule benötigt, da der EtherCAT-Slave automatisch erkennt, dass nach ihm kein weiterer Slave folgt.

Da dieser Zustand auch durch eine kurzzeitige Unterbrechung auftreten kann, wird auch im Kurzschlussfall versucht, ein Trägersignal bzw. das gerade im Durchlauf befindliche Telegramm über die Tx-Schnittstelle der Hin-Leitung zu senden, um eine wieder geschlossene Unterbrechung zu erkennen. Über die Rx-Schnittstelle der Hin-Leitung wird dann wieder ein Trägersignal erkannt und der Kurzschluss aufgehoben (Bild c).

Eine Unterbrechung kann natürlich auch auf der Hin-Leitung auftreten. Erkennt der EtherCAT-Slave an der Rx-Schnittstelle seiner Hin-Leitung kein Trägersignal mehr, so schließt er die Tx-Schnittstelle der Rückleitung mit der Rx-Schnittstelle der Hin-Leitung kurz. Über die Tx-Schnittstelle der Rückleitung wird in diesem Fall aber kein Trägersignal bzw. Telegramm mehr gesendet (Bild d).

Diagnose

Die Diagnosemöglichkeiten sind für den praktischen Einsatz eines verteilten Bussystems sicher ebenso entscheidend wie Performance-Daten, Topologie-Eigenschaften oder Verkabelungsaufwendungen. Auch hier erfüllt EtherCAT die Erwartungen an ein modernes Kommunikationssystem. Im Gegensatz zu Party-Line-Bussystemen (z.B. Profibus und CAN), bei denen die Teilnehmer alle am selben physikalischen Kabel hängen, und die Signale an alle Teilnehmer ohne Auffrischung gesendet werden, kommt bei Ethernet (zumindest ab 100 MBaud) und natürlich auch bei EtherCAT eine reine Punkt zu Punkt Übertragung zum Einsatz. Dadurch lassen sich Fehlerstellen oder auch nur sporadische Schwachstellen exakt lokalisieren, die sich bei Party-Line-Systemen entweder gar nicht oder nur mit speziellen Messaufbauten finden lassen.

Bruchstellen im logischen Kommunikationsring werden automatisch lokalisiert und geschlossen. Jeder Teilnehmer überwacht die Trägersignale sowohl auf der Hin- als auch auf der Rück-Richtung und kann entsprechende Störungen feststellen.

Sicherungsmaßnahmen

Bei EtherCAT wird (mittels Checksumme) geprüft, ob ein Telegramm korrekt übertragen und von allen adressierten Teilnehmern (mittels Working-Counter) korrekt

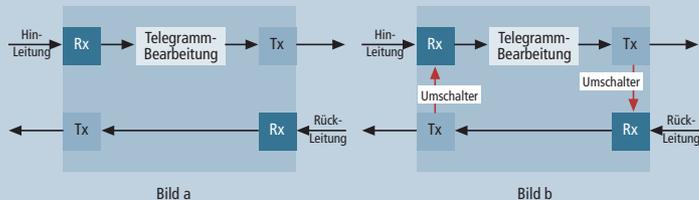


Bild a

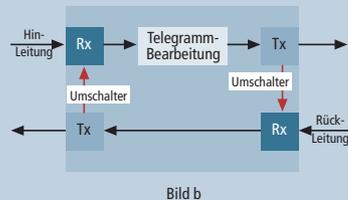


Bild b

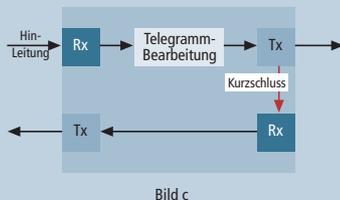


Bild c

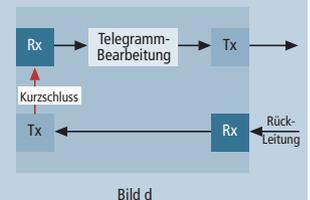


Bild d

Distributed-Clocks | Mit den Distributed-Clocks ist es bei EtherCAT möglich, in allen Busteilnehmern die gleiche Uhrzeit zu haben. Es gibt einen ausgewählten Teilnehmer, der die Master-Clock besitzt, auf die sich die Slave-Clocks der anderen Teilnehmer und der Steuerung synchronisieren. Dazu sendet die Steuerung in gewissen Abständen (so häufig, dass die Slave-Clocks innerhalb der

bearbeitet wurde. Als Checksumme wird die normale Ethernet-Checksumme benutzt, die sich am Ende des Ethernet-Telegramms befindet. Da ein oder mehrere Slaves das Telegramm während der Übertragung modifizieren, wird die Checksumme jedes Slaves neu berechnet. Bei einem erkannten Checksummenfehler wird ein Statusbit im EtherCAT-Slave gesetzt und ggf. ein Interrupt zum Master ausgelöst, so dass eine Fehlerstelle genau lokalisiert werden kann.

Bei einer Leseoperation werden die adressierten Daten vom Slave in das dafür vorgesehene Datenfeld eingefügt und beim Schreiben entsprechend entnommen. In beiden Fällen inkrementiert der adressierte Slave einen Working-Counter, der sich am Ende jedes EtherCAT-Kommandos befindet. Da der Master weiß, wie viele Slaves von dem Telegramm adressiert sind, kann er anhand des Working-Counters erkennen, ob alle Slaves ihre Daten korrekt ausgetauscht haben.

Protokoll

An einen Feldbus werden unterschiedliche Anforderungen gestellt, was die Übertragungseigenschaften von Daten betrifft. Parameterdaten werden azyklisch und in großen Mengen übertragen, wobei die zeitlichen Anforderungen relativ unkritisch sind, und die Übertragung in der Regel von der Steuerung angestoßen wird. Diagnosedaten werden ebenfalls azyklisch und ereignisgesteuert übertragen; die zeitlichen Anforderungen sind aber höher und die Übertragung wird in der Regel von einem Peripheriegerät angestoßen. Prozessdaten dagegen werden zyklisch mit verschiedenen Zykluszeiten übertragen. Dabei ist es wichtig, dass es nicht zu Zyklusausfällen kommt. Die zeitlichen Anforderungen sind bei der Prozessdatenkommunikation am höchsten.

Für die unterschiedlichen Kommunikationsarten gibt es bei EtherCAT entsprechende Adressierungsmöglichkeiten, die den Anforderungen jeweils optimal entsprechen.

Adressierung

Physikalische Adressierung: Mit der physikalischen Adressierung wird ein Teil des 64 kByte Adressraums eines Slaves gelesen oder geschrieben. Mit einem Telegramm ist immer genau ein Slave adressiert. Diese Adressierung wird besonders zur Übertragung von Parameterdaten genutzt. Zur Adressierung des Slaves gibt es zwei Varianten, die Auto-Increment- sowie die Fixed-Address-Adressierung.

Logische Adressierung: Bei der logischen Adressierung wird kein Slave einzeln sondern ein Teil eines 4 GByte großen logischen Adressraums angesprochen, der sich über beliebig viele Slaves verteilen kann. Die Zuordnung von physikalischen Adressen eines Slaves zu logischen Adressen im EtherCAT-Bus wird im Master projektiert und in der Startup-Phase an die Feldbus-Memory-Management-Units (FMMU) der Slaves übertragen. Eine FMMU hat die Aufgabe, den physikalischen Adressen eines Teilnehmers eine logische Adresse zuzuordnen. Je FMMU wird eine logische, bitorientierte Startadresse, eine physikalische Startadresse, eine Bit-Länge sowie ein Typ, der angibt in welcher Richtung gemappt werden soll (Inputs oder Outputs), konfiguriert. Damit ist es möglich, beliebige Daten eines EtherCAT-Slaves bitweise auf beliebige logische Adressen zu mappen. Der Slave überprüft beim Empfang eines Telegramms mit logischer Adressierung, ob eine seiner FMMUs einen Adressmatch hat, und fügt ggf. Daten an der entsprechenden Stelle des Datenfeldes in das Telegramm ein (Typ Inputs) bzw. entnimmt Daten von der entsprechenden Stelle aus dem Telegramm (Typ Outputs). Dadurch ist es möglich, Telegramme flexibel zusammenzustellen und an die Bedürfnisse der Steue-

renzen nicht auseinander laufen) ein spezielles Telegramm, in das der Busteilnehmer mit der Master-Clock seine aktuelle Uhrzeit einträgt. Das Telegramm wird dann von den Busteilnehmern mit Slave-Clock aus demselben Telegramm gelesen. Dies ist auf Grund der Ringstruktur von EtherCAT möglich, wenn die Master-Clock vor den Slave-Clocks angeordnet ist.

runge optimal anzupassen. Damit eignet sich diese Adressierungsart besonders für die Übertragung zyklischer Prozessdaten.

Multiple-Adressierung: Mit der Multiple-Adressierung können physikalische Adressbereiche mehrerer Slaves gelesen werden. Der erste Slave wird dabei per Stationsadresse adressiert und schaltet das Multiple-Read-Flag im Telegramm ein, über das die im Ring folgenden Slaves erkennen können, dass sie adressiert sind. Jeder Slave, der über den geforderten physikalischen Adressbereich verfügt, trägt seine Stationsadresse und die entsprechenden Daten in das Datenfeld ein, solange noch Platz ist.

Broadcast: Mit einem Broadcast-Write können physikalische Adressbereiche aller Slaves beschrieben werden; so lassen sich z. B. Zustandsübergänge bei allen Slaves gleichzeitig durchführen. Mit einem Broadcast-Read werden physikalische Adressbereiche aller Slaves verodert gelesen. Wenn eine Anwendung nur sinnvoll arbeiten kann, wenn sich alle Slaves im Okay-Zustand befinden, kann z. B. dieser Okay-Zustand sehr einfach und schnell mit einem Broadcast-Read von allen Teilnehmern abgefragt werden.

Querverkehr

Auch wenn EtherCAT ein klares Master-Slave-Kommunikationsmodell nutzt und damit ideal die hierarchische Steuerungstechnik unterstützt, lässt sich mit den Eigenschaften der Feldbus-Memory-Management-Unit (FMMU) sehr einfach der Querverkehr zwischen EtherCAT-Teilnehmern realisieren. Hierzu werden Speicherbereiche aus dem 4 GByte großen logischen Adressraum für Querverkehr reserviert und vom Master zyklisch ausgetauscht. Der Master stellt alternierend eine Leseanfrage und im nächsten Zyklus einen Schreibauftrag für den entsprechenden Speicherbereich. Alle entsprechend konfigurierten Teilnehmer blenden ihre Querverkehrsdaten ein bzw. entnehmen sie im nächsten Zyklus. Für den Master sind diese Daten transparent, und er sorgt nur für den zyklischen Austausch. Im Vergleich zu Partyline-Bussystemen, an denen alle Teilnehmer am selben Kommunikationsmedium hängen, vergeht zwar ein Zyklus; dies wird aber durch die überragende Nutzdatenrate und die dadurch möglichen kurzen Zykluszeiten mehr als wettgemacht. Die beschriebene Vorgehensweise hat auch den Vorteil, dass die Querverkehrsdaten von mehreren Quellen eingesammelt werden und dann im nächsten Zyklus bei allen angesprochenen Senken gleichzeitig ankommen. Bei einer Zykluszeit von z. B. 100 µs können bereits ca. 1.000 Bytes von fast beliebig vielen Quellen an ebenso viele Senken gesendet werden.

Ethernet over EtherCAT

Neben den bereits beschriebenen EtherCAT-Adressierungsarten, über die mit den EtherCAT-Teilnehmern kommuniziert wird, werden auch Standard-IP-basierte Protokolle wie TCP/IP, UDP/IP und alle darauf basierenden höheren Protokolle (HTTP, FTP, SNMP etc.) von einem Ethernet-Feldbus erwartet. Sinnvollerweise sollten dabei einzelne Ethernet-Telegramme transparent übertragen werden, da dadurch keine Einschränkungen bezüglich der zu übertragenden Protokolle entstehen. Prinzipiell gibt es zwei unterschiedliche Ansätze, um azyklische Ethernet-Telegramme im zyklischen Feldbusbetrieb zu übertragen. Die erste Variante besteht darin, im Zyklus eine entsprechende Zeitscheibe zu reservieren, in der die azyklischen Ethernet-Telegramme eingebettet werden können. Diese Zeitscheibe muss entsprechend groß gewählt werden, dass vollständige Ethernet-Telegramme darin Platz finden. Die übliche MTU (Maximum Transmission Unit) liegt bei

Feldbusanschlaltungen | Dank der Leistungsfähigkeit von EtherCAT können auch komplexe Busteilnehmer, wie z. B. ein Feldbusmastermodul, realisiert werden. In der Steuerung werden keine zusätzlichen Anschaltungen mehr benötigt. EtherCAT arbeitet quasi als PCI-Ersatz, in der Regel sogar mit Performance-Gewinn. Feldbusmasteranschlaltungen in Steuerungen haben eine Prozessabbild-

schnittstelle zum Austausch von zyklischen Prozessdaten und eine Mailbox-Schnittstelle zur Übergabe von azyklischen Parameter- oder Diagnosedaten. Das Prozessabbild kann über EtherCAT in der Regel mit einem einzigen Telegramm übertragen werden, wobei der Master das Output-Prozessabbild sendet und das Eingangsprozessabbild wieder empfängt.

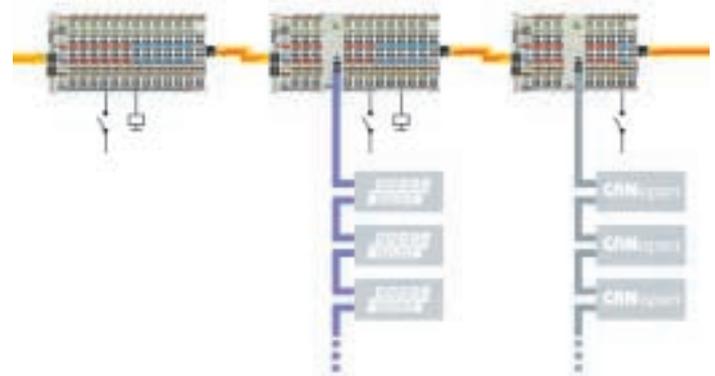
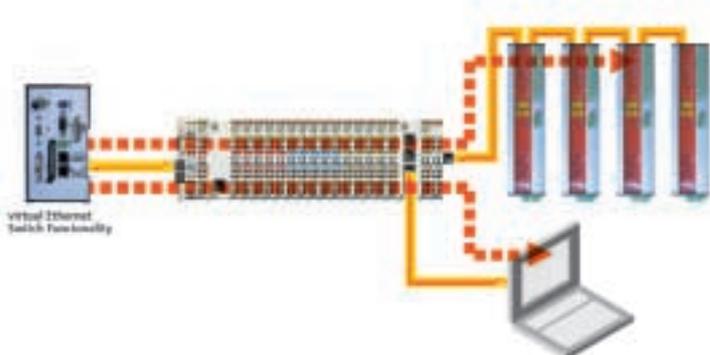
1.514 Byte, was in etwa 125 μ s bei 100 MBaud entspricht. Daraus ergibt sich für Systeme, die diese Variante nutzen, eine minimale Zykluszeit von etwa 200–250 μ s. Eine Verringerung der MTU führt häufig zu Problemen: Das IP-Protokoll lässt die Fragmentierung zwar prinzipiell zu; es wird aber davon abgeraten, und es wird sie in der nächsten Generation (IPv6) nicht mehr geben. Insbesondere bei UDP/IP-Übertragungen können zudem Datenkonsistenzprobleme auftreten. EtherCAT nutzt die zweite mögliche Variante, bei der Ethernet-Telegramme getunnelt und im entsprechenden Teilnehmer wieder zusammengestellt werden, bevor sie als vollständige Ethernet-Telegramme weitergeleitet werden. Dieses Verfahren schränkt die erreichbare Zykluszeit nicht ein, da je nach verfügbarer Bandbreite die Fragmente (EtherCAT statt IP-Fragmentierung) optimal eingestellt werden können. EtherCAT definiert dabei einen Standardkanal, der es prinzipiell jedem EtherCAT-Teilnehmer ermöglicht, am normalen Ethernet-Verkehr teilzuhaben. So kann zum Beispiel in einem intelligenten Antriebsregler, der mit 100 μ s Zykluszeit seine Prozessdaten austauscht, ein HTTP-Server integriert werden, der seine eigene Diagnoseschnittstelle in Form von Web-Seiten mitbringt.

Eine weitere Anwendung der übertragenen Ethernet-Telegramme realisieren die Hub- und Switchklemmen. Diese bieten an beliebiger Stelle im EtherCAT-System normale Ethernet-Ports mit entsprechenden RJ45-Buchsen, über die jedes Ethernet-Gerät angeschlossen werden kann. Dies kann z. B. ein Service-Rechner sein, der direkt mit der Steuerung kommuniziert, die Web-Seite eines intelligenten EtherCAT-Teilnehmers abfragt oder einfach über die Steuerung ins Intra- oder Internet routet. Die Switchklemme integriert zusätzlich zur Hubklemme einen Switchbaustein und bietet dadurch mehrere Ethernet-Anschlüsse in einem Gerät an. Im EtherCAT-Master ist ebenfalls eine Switchfunktionalität in Software integriert, die für das Routing der einzelnen Ethernet-Telegramme von und zu den EtherCAT-Teilnehmern und dem IP-Stack des Hostbetriebssystems zuständig ist. Die Switchfunktionalität ist dabei identisch mit der eines üblichen Layer-2-Switches und reagiert protokollunabhängig auf die verwendeten Ethernet-Adressen.

Fazit

In der Automatisierungstechnik zeichnet sich der Trend ab, Ethernet auch in der Feldebene einzusetzen. Unterschiedliche Ansätze versprechen hohe Bandbreiten, geringe Kosten, vereinfachte vertikale Integration, Einsatz von Standardkomponenten aus der Bürowelt und geringe Konfigurations- und Diagnoseaufwendungen. Das Ganze kombiniert mit der nötigen Echtzeitfähigkeit.

Ethernet over EtherCAT: Hub- und Switchklemmen



Spezielle EtherCAT-Teilnehmer: Feldbusmaster

Bei genauerem Hinsehen weichen viele der Argumente auf oder verkehren sich ins Gegenteil: Die vergleichsweise hohe Bandbreite von 100 MBaud Ethernet wird bei typischen I/O-Knoten mit wenigen Bytes Prozessdaten zunichte gemacht. Ein Teilnehmer, der z. B. vier Bytes pro Richtung hat, kommt auf eine Nutzdatenrate von 3–4 Prozent. Auch die Kostenseite spricht eher gegen den Einsatz im Feld. Neben den reinen Anschaltungskosten kommt noch ein recht hoher Bedarf an Rechenleistung für die Verarbeitung der Telegramme hinzu. Der Einsatz von Standardkomponenten hört meistens dann auf, wenn ein gewisses Maß an Echtzeitfähigkeit gefordert wird. Außerdem ist die typische geschichtete Sternverkabelung eher ungünstig für den Einsatz im Feld. Selbst die Konfiguration wird nicht leichter: Die Vergabe der notwendigen IP-Adressen wird nicht selten in Konflikten mit der IT-Abteilung enden.

EtherCAT geht hier andere Wege und verbindet die Vorteile aus der Feldbustechnik mit den ansonsten unbestreitbaren Vorteilen der Ethernet-Welt. Die zur Verfügung stehende Bandbreite wird nahezu vollständig ausgenutzt, die Kosten reduzieren sich auf eine einfache Asic-Anschaltung im EtherCAT-Teilnehmer. Standardkomponenten werden dort eingesetzt, wo sie auch wirklich Standard sind – in der Steuerung und nicht in der 2-Bit-I/O-Klemme. EtherCAT benötigt keine IP-Adressen, die Konfiguration läuft nach einfachen Algorithmen vom Master gesteuert automatisch ab. Die vertikale Integration ist aber trotzdem gegeben. Teilnehmer, die eine IP-Adresse haben möchten, bekommen diese auch und sind dann vollständig transparent im Netzwerk integriert.

Mit EtherCAT lassen sich hochperformante Maschinensteuerungen realisieren, bei denen viele verteilte Signale mit Zykluszeiten deutlich unter 100 μ s ausgetauscht werden können. Das System eignet sich aber ebenso gut für kostengünstige Steuerungsapplikationen, die mit drei Größenordnungen größeren Zykluszeiten auskommen, wie z. B. mit 100 ms in der Gebäudeautomatisierung. Jeder handelsübliche PC oder jeder Steuerungscontroller mit integriertem Ethernet-Port kann dabei als Master eingesetzt werden. EtherCAT bietet daher eine einheitliche, leistungsfähige Kommunikationsbasis für die gesamte Automatisierungstechnik. Von der „Klein“-SPS für weniger als 100 Euro bis zur Hochleistungs-CNC kann die gleiche Systemtechnik eingesetzt werden.

→ Die Langversion dieses Artikels finden Sie im Internet unter www.pc-control.net