

Sten Mückenheim

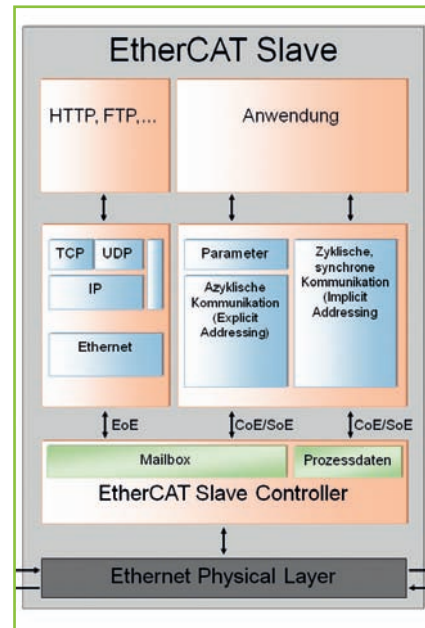
Feldbus trifft auf Ethernet

Mit Ethercat hat sich ein ethernetbasiertes Feldbus-System etabliert, welches sich unter anderem durch seine hohe Performance sowie eine nahezu unbeschränkte Netzausdehnung auszeichnet. Damit ist es prädestiniert für die Integration von bestehenden Feldbus-Protokollen und deren Funktionalitäten, wie das Beispiel „CAN application protocol over Ethercat“ – kurz CoE – zeigt.

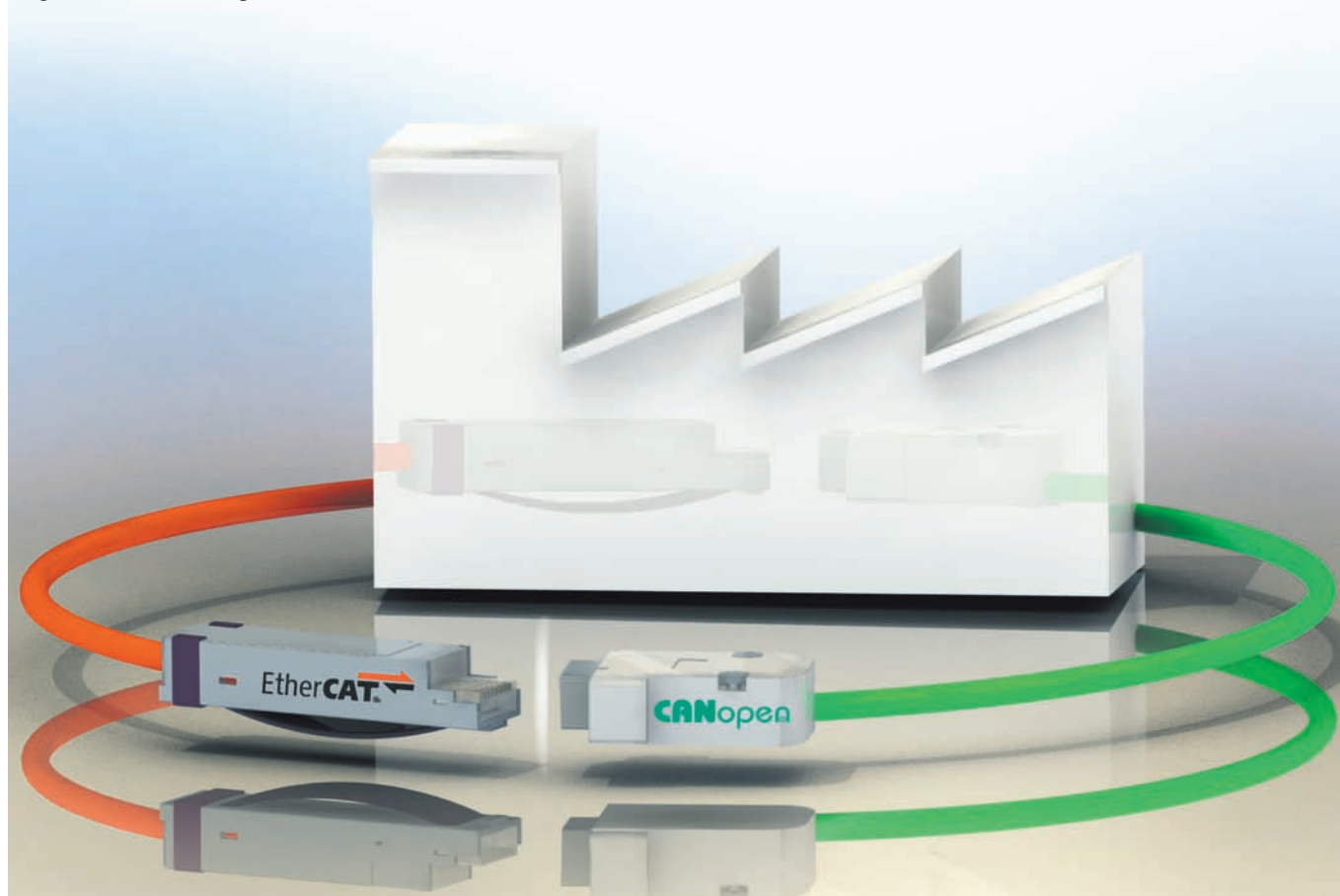
Die Anwender sind durch die langjährige Nutzung von herkömmlichen Feldbus-Systemen mit den bestehenden Geräteprofilen, deren Parameter und den dazugehörigen Software Tools vertraut. Aus diesem Grund wurde bei der Entwicklung von Ethercat (siehe *Kasten*, S. 48) auf die Bereitstellung eigener Geräteprofile für die unterschiedlichen Geräteklassen speziell verzichtet. Man geht vielmehr den Weg, einfache Schnittstellen zur Verfügung zu stellen, um den Anwendern die Migration von herkömmlichen Feldbussen zu Ethercat zu erleichtern.

Ein spezielles Beispiel hierfür ist die Migration von CANopen auf Ethercat.

Für eine Vielzahl von Geräteklassen stehen bereits CANopen-Geräteprofile zur Verfügung – etwa CiA-401 für E/A-Module oder CiA-402 für Antriebe. Bei Ethercat besteht die Möglichkeit, die gleichen Kommunikationsmechanismen, wie sie von CANopen her bekannt sind, zu nutzen. Zu nennen sind hier beispielsweise das Objektverzeichnis, Prozessdatenobjekte (PDO) oder Servicedatenobjekte (SDO). So lassen sich die Funktionalität von CANopen auf Geräten, auf denen Ethercat läuft, mit relativ geringem Aufwand implementieren und damit ein Großteil der bestehenden CANopen-Firmware der Geräte wieder verwenden. Des



Das Blockschaltbild zeigt den Aufbau eines Ethercat Slave – vom Physical Layer über den ESC (Ethercat-Slave-Controller) bis hin zur spezifischen Anwendung – basierend auf dem OSI-Referenzmodell.



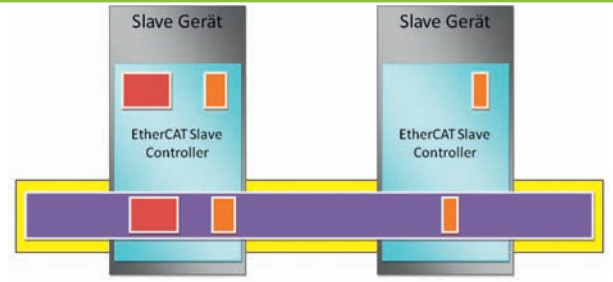
Ethercat – die Technologie

Ethercat (Ethernet for Control Automation Technology) ist ein ethernetbasiertes Feldbus-System, das von der Firma Beckhoff und der Ethercat Technology Group (ETG) entwickelt wurde und mittlerweile in den internationalen Standards IEC 61158, IEC 61784 sowie in ISO 15745-4 genormt ist.

Bei Ethercat wird ein Ethernet-Datenpaket nicht mehr in jedem Knoten zunächst empfangen, dann interpretiert und die Prozessdaten weiterkopiert. Vielmehr entnimmt jeder Slave die für ihn bestimmten Daten, während das Telegramm das Gerät durchläuft. Des Weiteren fügt das Slave-Modul seine Eingangsdaten während des Durchlaufs in das Telegramm ein. Dadurch werden die Telegramme nur wenige Nanosekunden verzögert. Auf diese Weise erreicht das ethernetbasierte Kommunikationssystem eine Nutzdatenrate von mehr als 90 %,

da ein Ethernet-Frame sowohl in Sende- als auch in Empfangsrichtung die Daten vieler Teilnehmer erreicht. Dabei werden die Vollduplex-Eigenschaften von 100Base-TX vollständig ausgenutzt, so dass effektive Datenraten von fast 200 MBit/s erzielbar sind. Die Update-Zeit für 1000 digitale I/Os beträgt nur 30 µs und für 200 analoge I/Os nur 50 µs – dies entspricht einer Sampling-Rate von 20 kHz. Hier inbegriffen ist bereits die I/O-Durchlaufzeit.

Die gesamte Abarbeitung des Protokolls von einem Slave erfolgt ausschließlich in Hardware und ist somit völlig unabhängig von der CPU-



Ethernet „on the fly“: In einem Durchlauf werden Ausgangsdaten aus dem Telegramm entnommen und Eingangsdaten eingefügt. Die Prozessdatengröße ist hierbei nahezu beliebig groß (1 Bit bis 60 KByte).

Performance, der Laufzeit vom Protokoll-Stack oder der Software-Implementierung. Es besteht die Möglichkeit, mit einem Ethernet-Frame bis zu 1486 Byte Prozessdaten auszutauschen. Das entspricht nahezu 12 000 digitalen Ein- und Ausgängen. Für die Übertragung dieser Datenmenge werden lediglich 300 µs benötigt.

Weiteren sind die Objekte optional erweiterbar, um der größeren Bandbreite, die Ethercat zur Verfügung stellt, Rechnung zu tragen.

Die Prozessdatenobjekte dienen dem schnellen und effizienten Austausch von Echtzeit-Daten, zum Beispiel von Soll- oder Istwerten. Im Ethercat-Telegramm selbst werden keine Objekte adressiert, sondern direkt die Inhalte der Prozessdaten von den zuvor gemappten Parametern gesendet. Die Servicedatenobjekte bilden den Kommunikationskanal für die Übertragung von Geräteparametern, beispielsweise bei der Programmierung der Geberauflösung. Da diese Parameter azyklisch – also nur einmal beim Hochfahren des Netzes – übertragen werden, haben die SDO-Objekte eine untergeordnete Priorität. Das SDO-Protokoll wird bei CoE direkt übernommen, so dass sich be-

stehende CANopen-Stacks nahezu ohne Änderung übernehmen lassen. Des Weiteren ist eine optionale Erweiterung definiert, die einerseits die 8-Byte-Beschränkung aufhebt und andererseits die vollständige Auslesbarkeit des Objektverzeichnisses ermöglicht.

Der Mailbox-Transfer

Ethercat setzt auf Multiprotokollfähigkeit und führt die unterschiedlichen Protokolle in einer einheitlichen Mailbox zusammen. CoE ist ein solcher Protokoll-Typ des so genannten Mailbox-Transfers. Dabei handelt es sich um die Standardvariante zum Austausch von Parameterdaten. Die Implementierung des Mailbox-Interfaces ist optional, aber zwingend notwendig, um azyklische Dienste nutzen zu können. Die Funktionalität des Mailbox-Transfers wird ebenfalls für weitere Protokoll-Typen, wie EoE (Ethernet over Ethercat), FoE (File Access over Ethercat), VoE (Vendor Specific Profil over

Ethercat) und SoE (Servo Drive over Ethercat) verwendet.

Für diesen Zweck sind zwei Sync-Manager (Speicherbereiche) reserviert:

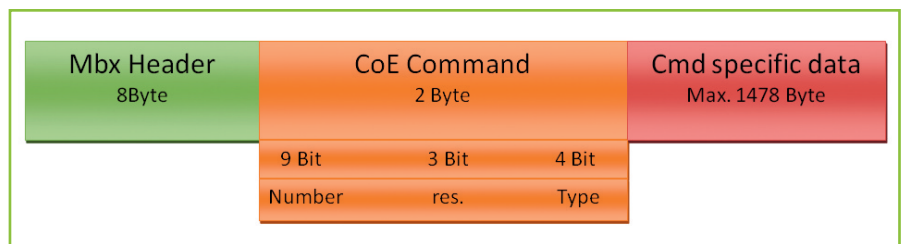
- Sync Manager 0 -> Master-to-Slave (MailBox In)
- Sync Manager 1 -> Slave-to-Master (MailBox Out)

Über den Bereich „Type“ im Mailbox-Header lässt sich identifizieren, um welchen Protokolltyp (EoE, CoE, SoE usw.) es sich handelt.

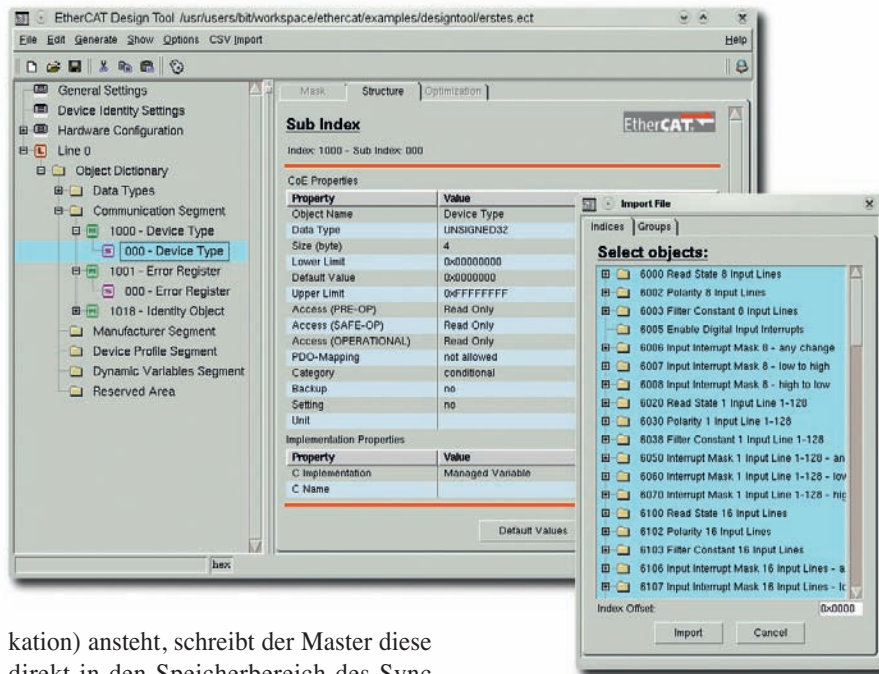
Bei CoE besteht die Möglichkeit, dass Ethercat-Geräte mit implementierter CoE-Funktionalität und CANopen-Geräte auf das gleiche Objektverzeichnis zugreifen. Für diesen Zweck ist der Bereich 0x1000 bis 0x1FFF im Objektverzeichnis reserviert. Somit wird die Kompatibilität zwischen CANopen und Ethercat garantiert. Wenn nun eine azyklische Kommunikation (Parameter, Diagnose, Geräte-Identifi-

Type	Message Type
0	reserved
1	Emergency Message
2	SDO Request
3	SDO Response
4	TxPDO
5	RxPDO
6	Remote Transmission request of TxPDO
7	Remote Transmission request of RxPDO
8	SDO-Information
9 – 15	Reserviert für künftige Mechanismen

Zuordnung der Kommunikationsmechanismen im CoE-Frame-Header



CoE-Frame, Mailbox Protocol: Das CoE-Frame ist als Datagramm in einem Ethercat-Frame eingebettet und besteht aus den drei Bereichen „Mbx Header“, „CoE Command“ und „Command specific data“.



Das Ethercat-Design-Tool zur Generierung des Objektverzeichnisses erleichtert den Einstieg in CoE und beschleunigt somit die Entwicklung von Endgeräten. Es unterstützt zudem Wartung und Diagnose durch übersichtliche Baumdarstellung aller Parameter und Daten.

innerhalb eines Projekts mehrere Hardwarekonfigurationen verwaltet werden können.

Kurzum: Mit dem Designtool steht ein Werkzeug zur Verfügung, welches den Entwickler von fehlerträchtigen und sich wiederholenden Tätigkeiten entlastet; zudem sichert es die Konsistenz von implementierter Funktionalität, die Gerätedokumentation und den Electronic Data Sheet (EDS, ESI). Im Grundumfang stellt das Tool Datenbanken mit dem Ethercat-Kommunikationsprofil zur Verfügung, optional auch diverse Geräteprofile. Das erzeugte Objektverzeichnis unterstützt zudem zahlreiche Optionen der Ethercat-Library von Port. *gh*

kation) ansteht, schreibt der Master diese direkt in den Speicherbereich des Sync Managers 0 (MailBox In). Der adressierte Slave entnimmt die Daten, verarbeitet sie und reagiert auf diese Anfrage. Er schreibt die Reaktion direkt in den Speicherbereich des Sync Managers 1 (MailBox Out). Mit dem nächsten Ethercat-Frame werden die azyklischen Daten zurück an den Master übertragen. Welcher Nachrichten-Typ bei diesem Kommunikationsvorgang übertragen wird, ist dem Bereich „Type“ im „CoE Cmd“ des CoE-Frames hinterlegt.

Um die Migration von CANopen-Geräteprofilen auf den ethernetbasierten Feld-

bus Ethercat zu erleichtern, hat beispielsweise die Firma Port ein spezielles Designtool entwickelt. Hierbei handelt es sich um ein Werkzeug zur Erstellung des Objektverzeichnisses für Ethercat-Applikationen. Es verwaltet Gerätedatenbanken, aus denen automatisch ein Objektverzeichnis, Konfigurations- und Initialisierungsdateien in C-Code, ein Electronic Data Sheet (EDS, ESI, XDD) und eine HTML-Dokumentation erzeugt werden. Weiterhin erleichtert das Tool die Konfiguration der Treiberpakete, wobei



Sten Mückenheim

ist Produktmanager OEM bei der Firma Port, Halle/Saale.