

# Topologievarianten von EtherCAT und deren Einfluss auf die Systemeigenschaften

Dipl.-Ing. (FH) Florian Häfele, EtherCAT Technology Group, Nürnberg

*In Maschinen der industriellen Automation werden heute eine Vielzahl von Aktoren und Sensoren einer oder mehrerer Steuerungen über ein oder mehrere Kommunikationssysteme verbunden, um die Steuerungsaufgabe effizient und flexibel realisieren zu können. Die physikalische Ausdehnung und Struktur der Maschine bestimmen dabei die Anforderungen an die Topologie des Kommunikationssystems. Hierzu erfolgt eine systematische Darstellung und Bewertung verschiedener Topologien sowohl aus Sicht des Endanwenders als auch aus Sicht des Geräteherstellers. Es werden sowohl typische als auch spezielle Systeme aus der Praxis untersucht und deren Topologieanforderungen dargestellt. Endanwendern wird eine Planungshilfe an die Hand gegeben; Geräteherstellern werden verschiedene Implementierungslösungen aufgezeigt.*

*EtherCAT / Topologie / Installation / Implementierung*

*Topology Variants of EtherCAT and their Influence on System Behaviour*

*Industrial automation systems connect a broad variety and number of actuators and sensors of one or more controllers via one or more communication systems by means of executing an efficient and flexible control task. System dimensions and structure dictate requirements to the communication system topology. In a systematic approach topology variants and features are described from users as well as from a developers view. Basic background information and specific solutions are covered. A system planning guide for end users is provided and implementation effort is estimated.*

*EtherCAT / Topology / Installation / Implementation*

## 1. EtherCAT-Funktionsprinzip

EtherCAT ist eine Ethernet-basierte Kommunikationstechnologie, die für die speziellen Anforderungen der modernen Automatisierungstechnik optimiert ist. Es werden Standard-Ethernet-Kabel verwendet, wie sie aus dem Office-Bereich bekannt sind. Der EtherCAT-Master, der das Netzwerk und die Kommunikation kontrolliert (EtherCAT ist ein Master-Slave-System), wird z. B. auf einem Industrie-PC in Software realisiert; einzige Hardwareanforderung ist eine Standardnetzwerkkarte.

Die EtherCAT-Slave-Geräte basieren auf einem speziellen EtherCAT-Kommunikationschip, dem sogenannten EtherCAT-Slave-Controller, kurz ESC. In diesem wird die gesamte Prozessdatenkommunikation abgearbeitet. Dadurch wird eine einheitliche Übertragungsgeschwindigkeit und Zuverlässigkeit der Kommunikation garantiert, unabhängig von der jeweiligen gerätespezifischen Implementierung.

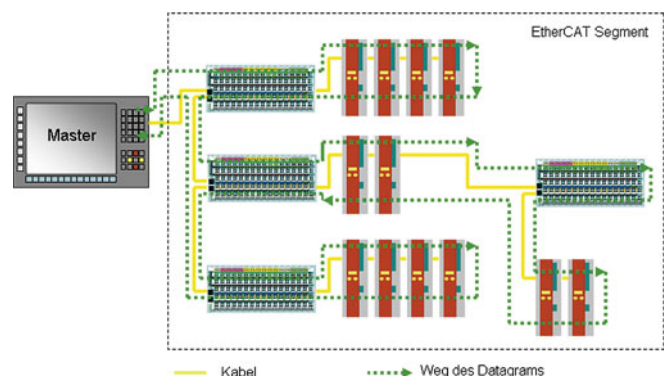
Über gängige Prozessdatenschnittstellen mit einem DPRAM werden die Prozessdaten zwischen EtherCAT und dem Applikations-Controller ausgetauscht.

Die Topologie des Netzwerks kann vom EtherCAT-Netzwerkconfigurator online ermittelt werden, indem Portanzahl und Link-Status jedes einzelnen Gerätes ausgelesen werden.

Der Adressraum bietet die Möglichkeit zur Adressierung von über 65 000 Geräten in einem Segment. Der Betriebszustand aller Geräte wird in jedem Zyklus überprüft; Gerätefehler werden zyklussynchron erkannt, so dass das Anwenderprogramm in Echtzeit reagieren kann.

### 1.1 Verarbeitung im Durchlauf

EtherCAT-Datagramme werden im Durchlauf bearbeitet. Dabei werden Lese- und Schreibzugriffe immer nur auf einem kleinen Ausschnitt des gesamten Telegramms ausge-



**Bild1: Datagramm-Bearbeitung im Durchlauf.**

führt. Das Telegramm wird sofort zum nächsten EtherCAT-Gerät weitergeleitet (Bild 1) und nicht zunächst empfangen, dann bearbeitet und danach versendet.

### 1.2 Bearbeitungsreihenfolge

EtherCAT arbeitet im Full-Duplex-Mode. Telegramme werden auf einem Leitungspaar in der „Processing Direction“, also in Richtung vom Master zum Slave, versendet. Die Frames werden vom EtherCAT-Gerät nur in dieser Richtung bearbeitet und zum nachfolgenden Gerät weitergeleitet, bis das Telegramm alle Geräte durchlaufen hat. Das letzte Gerät sendet das Telegramm auf dem zweiten Leitungspaar im Kabel in „Forwarding Direction“ zurück zum Master. Dabei bildet EtherCAT immer eine logische Ringstruktur, unabhängig von der gewählten Topologie (siehe Bild 1).

### 1.3 Einheitliche Systemzeit und Synchronisierung

EtherCAT-Geräte implementieren eine hochpräzise Zeit in Hardware, genauer im ESC. Diese verteilten Uhren geben dem EtherCAT-Synchronisierungsmechanismus seinen Namen „Distributed Clocks“ (DC). In der Regel wird das erste DC-Gerät nach dem Master als Referenzuhr verwendet, auf das alle anderen Geräte abgeglichen werden. Hierzu zählt der Ausgleich der unterschiedlichen Startzeiten der Uhren, einschließlich der des Masters, und der Verzögerung durch Kabel und Hardware. Auf der so erzeugten einheitlichen Zeitbasis können Anwendungen, die Daten auf mehreren Geräten gleichzeitig und synchron ausgeben oder einlesen sollen, realisiert werden. Hochgenauen Antriebs- oder Mess-technikanwendungen steht mit diesem Mechanismus eine Zeitbasis mit einer Abweichung von weit unter 1 µs zur Verfügung.

### 1.4 Übertragungsphysik und Signalerzeugung

EtherCAT wird auf den Medien 100BASE-TX, 100BASE-FX und EBUS übertragen. EBUS, das auf der LVDS-Übertragungsphysik basiert, wird für die geräteinterne Kommunikation verwendet und lässt sich kompakt und kostengünstig implementieren.

Die 100BASE-Codiervarianten werden über PHYs an den ESC angebunden. Als Schnittstelle dient das MII (Media Independent Interface).

Jeder ESC erzeugt das physikalische Signal neu, wodurch eine gleichbleibende, topologieunabhängige Signalqualität erreicht wird. Außerdem ist dadurch eine unbegrenzte Zahl von Medienwechseln möglich.

## 2. Topologiemöglichkeiten mit EtherCAT

Topologien können sich in ihrer Komplexität erheblich unterscheiden. Die Bandbreite reicht über einfache Topologien, wie Linie, Baum, Stern oder gemischte Varianten, bis zu solchen mit segment- und systemüberschreitendem Daten- und Zeitstempelaustausch.

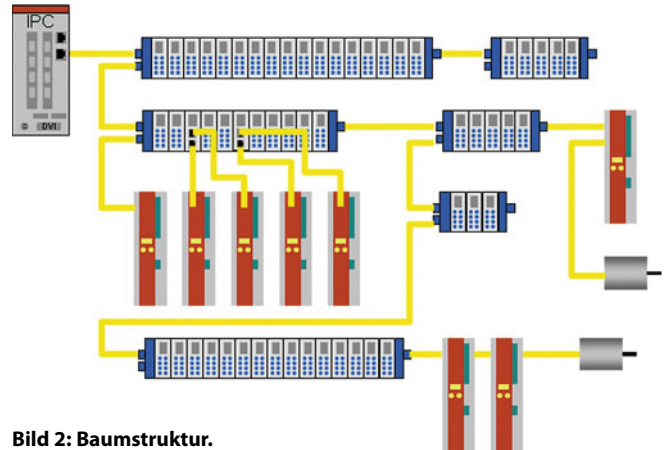


Bild 2: Baumstruktur.

### 2.1 Baumstruktur

Die Baumstruktur kombiniert die Topologie „Daisy Chain“ mit Stichleitungen/Linientopologie. Diese klassischen Topologien werden sowohl in reiner Form als auch beliebig kombiniert von EtherCAT unterstützt.

EtherCAT-Geräte unterscheiden sich in der Port-Anzahl. Geräte mit mehr als zwei Ports (viele ESC unterstützen bis zu vier Ports) dienen zum Anschluss von Stichleitungen (Bild 2).

### 2.2 Sterntopologie

Die Sterntopologie wird mit EtherCAT-Abzweigen umgesetzt. Sie hat – wie auch die Baumtopologie – den Vorteil, dass ein Geräteausfall oder ein Leitungsbruch nicht zum Abkoppeln anderer Geräte führt (siehe Bild 3). Auch bei dieser Topologie bleibt der logische Ring erhalten und damit die darauf basierenden Echtzeiteigenschaften.

### 2.3 Hot-Connect

Hot-Connect erlaubt das Ab- und Ankoppeln von Geräten oder Segmenten im laufenden Betrieb. Besonders komfortabel geht dies mit Geräten mit zusätzlichem ID-Switch: Diese werden unabhängig von ihrer Position im Netz erkannt und können daher an jedem freien Port angeschlossen werden. Dank der Eigenschaften des EtherCAT-Slave-Controllers wird

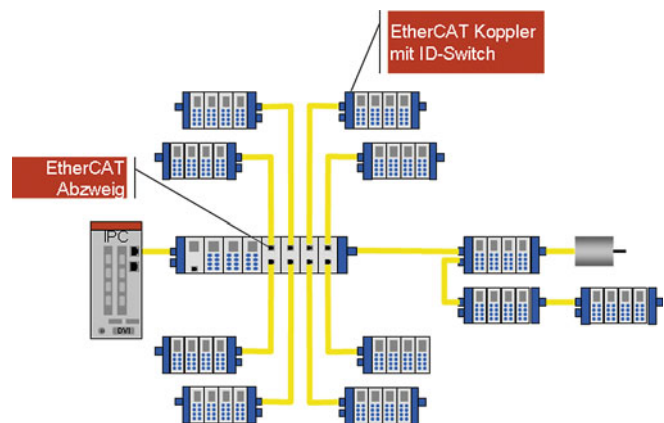


Bild 3: Sterntopologie mit Echtzeit.

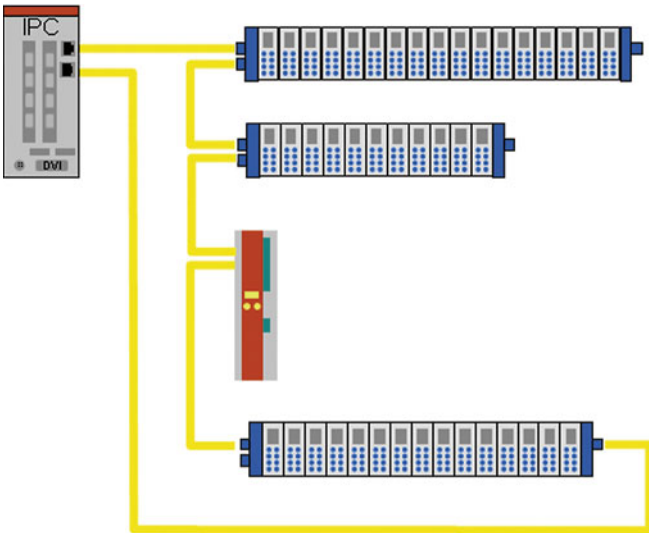


Bild 4: Ringstruktur für Kabelredundanz.

der Ankoppelvorgang sehr schnell erkannt. Ports können auch vom Master aus gezielt abgeschaltet werden, bevor das Gerät oder Segment abgekoppelt wird.

## 2.4 Ringstruktur für Kabelredundanz

Für die Umsetzung der Kabelredundanz wird die Ringtopologie verwendet (siehe Bild 4). Hierzu wird das letzte Gerät in der Bearbeitungsreihenfolge mit dem Master verbunden. Als letztes Gerät eignen sich alle Geräte mit mindestens einem freien MII-Port. Für die Redundanz wird hardwareseitig lediglich ein zweiter Ethernet-Port verwendet; der Master bleibt ansonsten auch hier eine Software-Implementierung.

## 2.5 Synchronisierung mehrere EtherCAT-Netze

Der Datenaustausch zwischen zwei oder mehreren EtherCAT-Netzen kann sehr einfach mit mehreren Switchports oder einer Bridge realisiert werden (Bild 5). Dabei können zwei Switchports aus unterschiedlichen Segmenten miteinander verbunden werden um Daten auszutauschen.

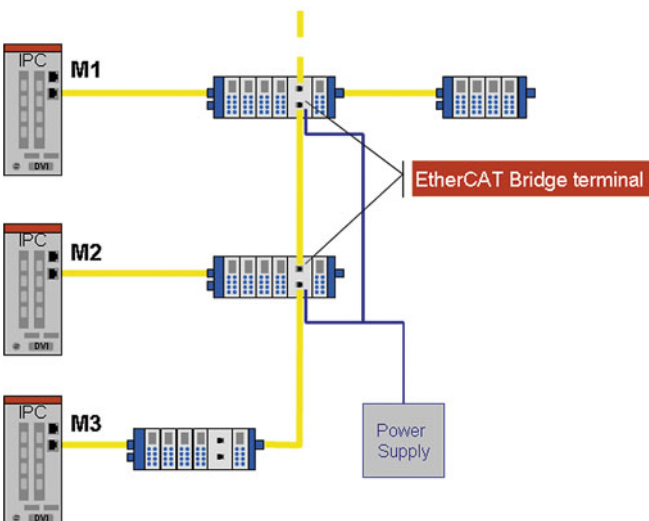


Bild 5: Synchronisierung mehrerer EtherCAT-Segmente.

Mit einem Bridge-Gerät können zusätzlich zum Datenaustausch Netzwerke synchronisiert werden, um damit über Systemgrenzen hinweg eine einheitliche Zeitbasis zur Verfügung zu stellen. Dies ist zum Beispiel im Prüfstandsbau oder bei modularen Maschinen mit mehreren Steuerungen von besonderem Interesse.

EtherCAT bietet neben dieser auch noch weitere Möglichkeiten der Master-Master-Kommunikation.

## 2.6 Master-Master-Kommunikation

Master können z.B. direkt mit Standard-Switchen und einer zweiten Netzwerkkarte oder alternativ über ein EtherCAT-Switchport-Gerät verbunden werden, um untereinander sowohl azyklische als auch zyklische Daten auszutauschen.

## 2.7 Slave-Slave-Kommunikation

Beim Austausch von Daten zwischen verschiedenen Slaves übernimmt der Master die Funktion eines Routers. Daten werden von einem Gerät gelesen und im Master vom Eingangsprozessabbild ins Ausgangsprozessabbild kopiert und anschließend versendet. Dies kann sogar in ein und demselben Steuerungszyklus geschehen.

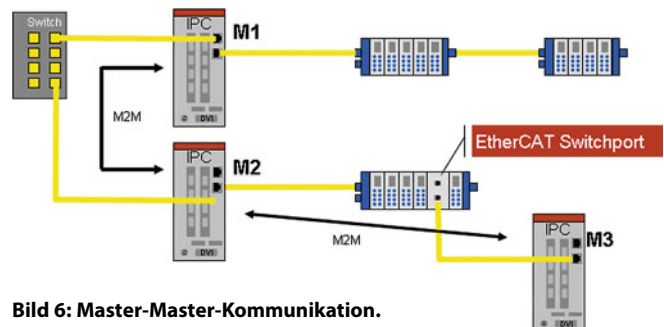


Bild 6: Master-Master-Kommunikation.

Für höchste Anforderungen kann auch die topologieabhängige Variante der Slave-Slave-Kommunikation eingesetzt werden: Ein Gerät fügt Daten in das durchlaufende Telegramm ein, die von nachfolgenden Geräten ausgewertet werden.

## 3. Planungshilfe für Endanwender

Im Folgenden werden einige der wichtigsten Punkte für die Netzwerkplanung erläutert.

### 3.1 Bestimmung der Durchleitezeit

Da EtherCAT das Full-Duplex-Verfahren nutzt, können mehrere Telegramme hintereinander gesendet werden, ohne auf die Rückkehr des vorangegangenen Frames zu warten. Daher ist die Zykluszeit nicht mit der Durchleitezeit gleichzusetzen. In der Praxis wird diese jedoch oft als minimale Zykluszeit angesetzt; sie kann unter Vernachlässigung aller Opti-

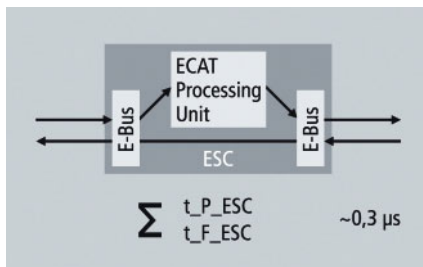
mierungsmöglichkeiten und der Symmetrie der Ein- und Ausgangsdaten folgendermaßen bestimmt werden:

$$t_{\text{Delay}} = m \cdot t_{\text{EBUS}} + n \cdot t_{\text{MII}} + t_{\text{PD}} + 2 \cdot t_{\text{Kabel}}$$

Verzögerung	Beschreibung
$m \cdot t_{\text{EBUS}}$	Verzögerung durch $m$ Geräte mit 2 EBUS-Ports
$n \cdot t_{\text{MII}}$	Verzögerung durch $n$ Geräte mit 2 MII-Ports
$t_{\text{PD}}$	+ Verzögerung durch die Prozessdatenlänge (Outputs + Inputs) bei 100Mbit/s (hierbei wird vernachlässigt, dass sich bei symmetrischem Verhältnis von I/O Daten pro Gerät die Prozessdatenlänge im Frame halbiert)
+ 26 Byte Overhead pro Ethernet-Frame	
+ 12 Byte Overhead/Datagramm	
$t_{\text{Kabel}}$	Verzögerung durch 100BASE-TX-Kabel (~ 5 ns/m).
$t_{\text{P}}$	Verzögerung in Processing Direction
$t_{\text{F}}$	Verzögerung in Forwarding Direction

### 3.1.1 Verzögerung durch ein Gerät mit 2 EBUS-Ports ( $t_{\text{EBUS}}$ )

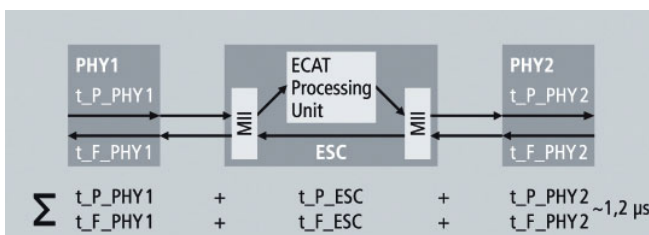
Die Verzögerung durch ein EtherCAT-Gerät mit zwei EBUS-Ports (z.B. modulare I/Os), wird durch die Hardwareverzögerung des ESCs bestimmt und beträgt etwa 0,3  $\mu\text{s}$ .



**Bild 7:** Verzögerung eines EBUS-Gerätes.

### 3.1.2 Verzögerung durch ein Gerät mit 2 MII-Ports ( $t_{\text{MII}}$ )

Die Verzögerung durch ein EtherCAT-Gerät mit 2 MII-Ports (z.B. Antrieb) mit zwei RJ45-Ethernet-Steckern wird durch die Hardwareverzögerung des ESCs und der zwei PHYs bestimmt. Sie beträgt – je nach PHY – etwa 1,2  $\mu\text{s}$ .



**Bild 8:** Verzögerung eines MII-Geräts.

## 3.2 Auswahl der Topologie

Die Auswahl der Topologie hat bei EtherCAT keinerlei nachteiligen Einfluss auf die Funktionalität, Echtzeit oder andere Features. Diese Tatsache ermöglicht es, die Topologie voll auf die physikalische Ausdehnung der Anlage und die Verwendung besonderer Funktionalitäten, wie Hot-Connect oder Redundanz, zuzuschneiden.

## 3.3 Einsatz von Hot-Connect

Viele Applikationen erfordern eine Änderung der I/O-Konfiguration während des Betriebes. Beispiele sind Bearbeitungszentren mit wechselnden, sensorbestückten Werkzeugsystemen oder Druckmaschinen, bei denen einzelne Druckwerke abgeschaltet werden. Die Protokollstruktur des EtherCAT-Systems trägt diesen Anforderungen Rechnung: Die Hot-Connect-Funktion erlaubt es, Teile des Netzwerkes im laufenden Betrieb an- und abzukoppeln, umzukonfigurieren und so flexibel auf wechselnde Ausbaustufen zu reagieren (siehe auch Bild 3).

## 3.4 Einsatz von Hot-Swap

Kommt es bei komplexen Geräten zum Austauschfall, muss das Ersatzgerät identisch parametrierbar sein. Dies erfordert Kenntnisse im Umgang mit speziellen Parametrierungsprogrammen sowie über die einzustellenden Parameter. Dieses Wissen ist oft vor Ort nicht vorhanden.

Bei EtherCAT kann dieses Problem durch die Hot-Swap-Funktionalität umgangen werden. Dabei werden die Parameterdaten im Master gespeichert und automatisch beim Einschalten auf das Gerät gespielt.

## 3.5 Einsatz von Distributed-Clocks

Die Funktionalität des Synchronisierungsmechanismus Distributed-Clocks (DC) ist unabhängig von der Netzwerkstruktur. Geräte mit und ohne DCs können beliebig angeordnet werden. Die Synchronisierung der Uhren wird automatisch vom EtherCAT-Master durchgeführt, so dass der Anwender keine speziellen Einstellungen vornehmen muss.

## 4. Implementierungsaspekte

Für den Gerätehersteller stellt sich bei all den aufgezeigten Möglichkeiten, die EtherCAT bietet, die Frage nach dem damit verbundenen Entwicklungsaufwand. Manche Features werden vom ESC automatisch unterstützt, andere erfordern Software- und/oder Hardwareerweiterungen.

### 4.1 Port-Auswahl

Eine Haupteigenschaft eines EtherCAT-Gerätes ist die Portanzahl und der Porttyp. Um die Flexibilität der Topologie zu erhalten, sollte ein EtherCAT-Gerät mindestens zwei Ports besitzen, so dass weitere Geräte in Reihe angeschlossen werden können. Für nichtmodulare Geräte, wie einen Antrieb,

bedeutet dies mindestens zwei MII-Ports, bei modularen Geräten können auch zwei EBUS-Ports verwendet werden.

Geräte mit zusätzlichen Infrastruktureigenschaften, wie z. B. eine Verteilerklemme für Sterntopologie oder ein Physikumsitzer von 100BASE auf EBUS und umgekehrt, können auch mehrere verschiedene Ports haben. Portanzahl und Porttyp spielen eine entscheidende Rolle bei der Auswahl des ESCs, da diese sich hierin deutlich unterscheiden.

### 4.2 Hot-Connect

Für Hot-Connect empfiehlt es sich, eine positionsunabhängige Adresse zu unterstützen, damit ein Maschinenmodul an jedem beliebigen freien Port innerhalb des Netzwerks angeschlossen und eindeutig identifiziert werden kann. Diese Adresse muss auch nach einem Spannungsverlust noch zur Verfügung stehen.

EtherCAT verwendet für Hot-Connect eine zweite Adresse, den sogenannten „Station-Alias“. Dieser kann je nach Implementierung aus einem ESC-Register ausgelesen oder als Prozessdatum übertragen werden. In beiden Fällen wird eine Einstellmöglichkeit auf dem Slave-Gerät benötigt, die zum Beispiel über einen DIP-Switch oder ein Bedienfeld realisiert werden kann. Eine andere Quelle für den Station-Alias kann der nichtflüchtige Konfigurationsspeicher des ESC sein, aus dem die Adresse in ein Register geladen wird.

### 4.3 Hot-Swap

Wird im Ersatzfall der Gerätetausch notwendig, muss bei komplexen Geräten wie Antrieben oftmals eine Neuparametrierung vorgenommen werden. Um dies zu vermeiden, beschreibt EtherCAT ein Verfahren für Backup-Parameter. Dieses ermöglicht es dem Gerätehersteller, wichtige Parameter zu kennzeichnen. Alle gekennzeichneten Parameter werden nach dem Austausch automatisch auf das neue Gerät geladen.

Im Slave muss hierzu eine Erweiterung der Firmware erfolgen, um den Backup-Mechanismus zu unterstützen. Dazu gehört die Erweiterung zur Kennzeichnung der Backup-Objekte, das Abspeichern dieser in nichtflüchtigen Speicher und die Überprüfung auf Gültigkeit der Daten.

### 4.4 Redundanz

Für die Kabelredundanz wird das EtherCAT-Telegramm auf zwei Netzwerkports versendet. Im Normalfall wird nur das Telegramm in Processing Direction bearbeitet, während das zweite Telegramm in Forwarding Direction das Netzwerk durchläuft und nicht bearbeitet wird. Im Redundanzfall durchlaufen beide Telegramme einen Teil des Netzwerkes in Processing und in Forwarding Direction.

Im Slave erfordert das Redundanzkonzept keine Erweiterung. Die benötigte Funktionalität basiert auf vorhandenen Mechanismen, die jeder ESC unterstützt.

Auf Master-Seite ist eine Software-Erweiterung notwendig, die beide versendeten Telegramme auswertet und ein vollständiges Bild der Kommunikation für die Master-Applikation erzeugt.

## Zusammenfassung

EtherCAT bietet die Möglichkeit verschiedene Topologien (z.B. Linie, Daisy Chain, Baum, Stern) beliebig zu kombinieren. Dabei bleiben Echtzeitfähigkeit und hochgenaue Synchronisierung uneingeschränkt erhalten. Besondere Features wie Hot Connect, Hot Swap oder Redundanz sind sowohl in der Implementierung als auch in der Anwendung einfach und schnell umzusetzen. Der logische Ring, der den Weg eines EtherCAT-Frames im Netzwerk immer beschreibt, ermöglicht es dem Endanwender, die Durchleitezeit schnelle zu Bestimmung.

## Literatur

- [1] ANSI/TIA/EIA-644 "Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits".
- [2] IEC 61158-2 (Ed.4.0): Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition.
- [3] IEC 61158-3/4/5/6-12 (Ed.1.0): Industrial communication networks – Fieldbus specifications - Part 3-12: Data-link layer service definition – Part 4-12: Data-link layer protocol specification – Part 5-12: Application layer service definition – Part 6-12: Application layer protocol specification -Type 12 elements (EtherCAT).
- [4] EtherCAT Technology Group website, <http://www.ethercat.org>
- [5] ETG.1020: "EtherCAT Guidelines and Protocol Enhancements", EtherCAT Technology Group.

Manuskripteingang: 8.10.08



Dipl.-Ing. (FH) *Florian Häfele* (28) hat Feinwerktechnik und Automatisierungstechnik studiert. Seit 2006 ist er bei der EtherCAT Technology Group für Implementierungssupport und Schulung zuständig und unterstützt die Weiterentwicklung und Standardisierung der EtherCAT-Technologie, zu der auch der Aufbau und die Leitung des EtherCAT-Conformance-Test-Labors zählt.

Adresse: EtherCAT Technology Group, Ostendstr. 196, D-90482 Nürnberg, Tel. +49 911 5405614, Fax +49 911 5405629, E-mail: [f.haeefe@ethercat.org](mailto:f.haeefe@ethercat.org), Internet: [www.ethercat.org](http://www.ethercat.org)