

EtherCAT und TSN: die perfekte Ergänzung

Was ist TSN?

TSN (Time-Sensitive Networking) hat zum Ziel, die Echtzeiteigenschaften (Synchronisation, Verfügbarkeit, garantierte Ressourcen und schnelle Reaktion) der heutigen kommerziellen (IEEE 802.1) Netze zu verbessern. TSN ist eine Task Group innerhalb der IEEE 802.1 Working Group, die eine Serie von Switch-Standards entwickelt, welche optional zu den bisherigen Funktionen in Switches und Endgeräten unterstützt werden können.

Die zeitkritische Kommunikation läuft parallel zur klassischen Kommunikation nach dem Best-Effort-Prinzip, bei dem der Datenaustausch so schnell wie möglich, abhängig von der Verkehrslage, funktioniert.

Wie diese Kommunikationstechnologien mit den Anwendungsprozessen zu kombinieren sind, lässt der TSN-Standard offen.

Wie funktioniert TSN?

Anders als üblich wird bei TSN nicht beliebig kommuniziert, sondern über vorher zu etablierende Datenströme (Streams). Ein Stream ist ein Datenkanal mit definierter Bandbreite von einem Sender (Talker) zu einer Gruppe von Empfängern (Listener). Die Streams werden meist vom Netz gebündelt. Diese Bündel (Traffic Classes) werden von den Netzkomponenten bevorzugt behandelt.

Eine besondere Stärke von TSN ist die Synchronisation der Netzteilnehmer. Damit ist es möglich, koordinierte Aktionen sehr präzise durchzuführen.

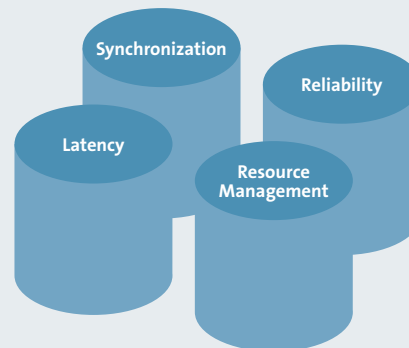
EtherCAT-Kommunikation mit TSN

Mit TSN kann man über ein IEEE 802.1 Netz mit kalkulierbarer Verzögerung kommunizieren. So kann man sehr flexibel mit einer einzigen hochleistungsfähigen Ethernet-Schnittstelle Automatisierungsaufgaben lösen:

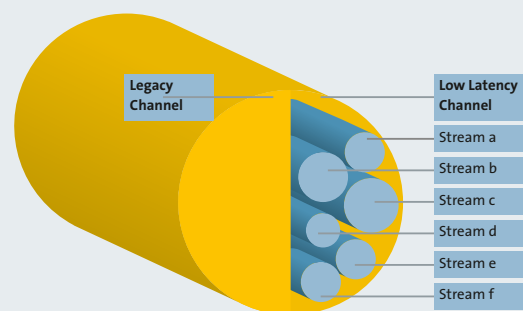
- mehrere Slave-Segmente betreiben
- mit einer überlagerten Steuerung kommunizieren
- Video-Kameras anschließen
- ...

Eine Gruppe von EtherCAT-Teilnehmern wird dabei als ein Segment an die TSN-Domäne angebunden, welche aus EtherCAT-Sicht wie ein großer Switch zu betrachten ist. Die Anpassung erfolgt über eine Adaptionsschicht. Diese kann im Master mittels Softwareschicht realisiert werden, sowie im Switch oder im ersten Slave des EtherCAT-Segments. Die EtherCAT-Slaves bleiben von dieser Funktionalität unberührt.

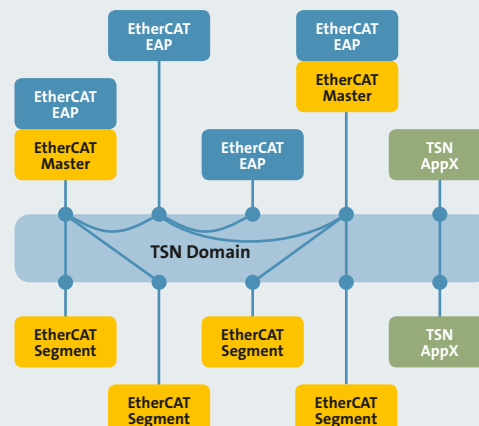
Wesentliche Elemente von TSN



TSN-Kommunikation über Streams



Beispiel eines EtherCAT-TSN-Netzwerks



EtherCAT und TSN: die perfekte Ergänzung

Kombinationen mit Hilfe von TSN

Natürlich ist die Trennung von maschineninterner Kommunikation von der überlagerten Kommunikation wie bisher möglich. Die zellenübergreifenden Interaktionen über TSN können mit deutlich verbesserter Präzision im Vergleich zu herkömmlichen geschichteten Netzwerken stattfinden.

EtherCAT Master-Slave Kommunikation kann über TSN betrieben werden. Der EtherCAT-Master-Code bleibt unverändert. Im Ethernet-Treiber wird eine Multiplex-Schicht eingebaut, die den Betrieb mehrerer EtherCAT-Kanäle und auch den Zugang weiterer Applikationen erlaubt.

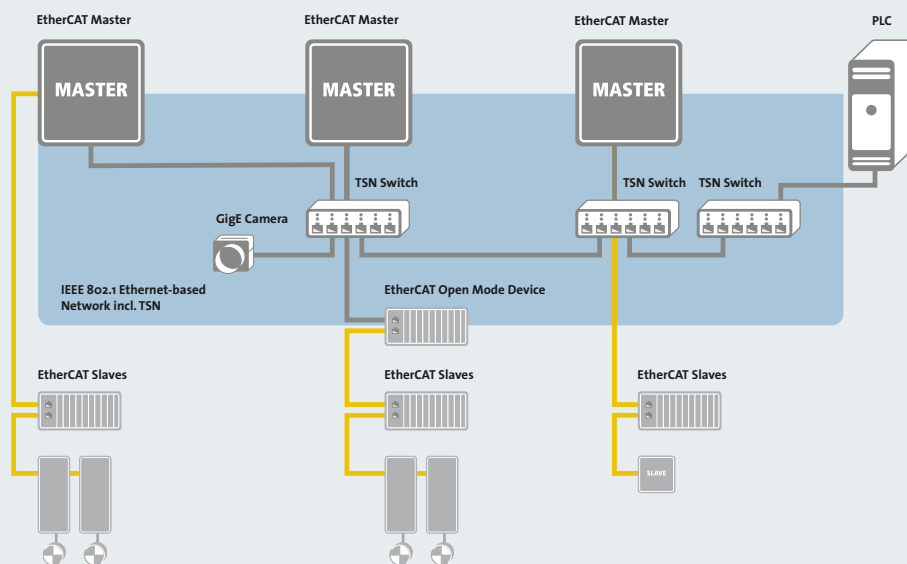
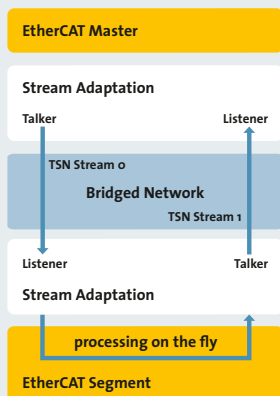
Unabhängig der eigentlichen TSN-Funktionen der Switches wird eine **Adaption zwischen TSN und EtherCAT** angeboten, entweder integriert in ein EtherCAT-Slavegerät, als eigenständiges Gerät, oder integriert in einen Switch.

Der TSN-Backbone kann aus einer Reihe von Komponenten unterschiedlicher Hersteller bestehen (z. B. Hilscher, Hirschmann-Belden, Moxa, Xilinx/SoC-e). Weitere TSN-Komponenten können ergänzt werden, wenn sie das in der EtherCAT TSN Profildefinition ETG.1700 geforderte TSN-Subset unterstützen.

EtherCAT TSN Profil

- Integration in heterogene Netze
- keine Veränderung im EtherCAT Slave
- einfache Umsetzung im Master
- einfaches Handling in TSN durch effiziente Telegramm-Nutzung
- Liaison mit IEEE 802.1

Stream adaptation



Was ist EtherCAT?

EtherCAT (Ethernet for Control Automation Technology) ist eine offene Industrial-Ethernet-Lösung. EtherCAT setzt neue Maßstäbe in Bezug auf Echtzeit-Performance, niedrige Kosten, flexible Topologie sowie einfache Handhabung.

www.ethercat.org

EtherCAT
Technology Group

EtherCAT Technology Group

ETG Headquarters
Ostendstraße 196
90482 Nürnberg, Germany
Telefon: + 49 911 540 56-20
E-Mail: info@ethercat.org