

EtherCAT-Implementierung: Möglichkeiten, Schnittstellen, Aufwand

Dipl.-Ing. Michael Jost, Beckhoff Industrie Elektronik Verl
Dipl.-Ing. Martin Rostan, Beckhoff Industrie Elektronik Nürnberg

EtherCAT Implementation: Options, Interfaces, Effort

EtherCAT is a high performance real-time Ethernet solution. Besides its outstanding performance data it provides various process data interfaces and is easy to implement. The basic functionality, the hardware and software effort necessary to develop an EtherCAT Master or Slave Device are described here.

Keynotes: EtherCAT, Industrial Ethernet, Implementation, Real-time

1 Einführung: Kurzübersicht EtherCAT Technologie

1.1 Ethernet auch auf I/O Ebene

Feldbusse sind aus der Automatisierungstechnik nicht mehr wegzudenken. Sie haben sich bewährt und sind etabliert. Erst durch die Feldbus-Technik wurde der Einsatz der PC-basierten Steuerungen auf breiter Front sinnvoll möglich. Während die Steuerungs-CPU's – speziell bei IPCs – rasant an Performance gewinnen, wirken die herkömmlichen Feldbussysteme zunehmend als „Flaschenhals“ und begrenzen die erreichbare Leistungsfähigkeit der Steuerungssysteme.

Oberhalb der Feldbussysteme, zur Vernetzung von Steuerungen, ist Ethernet schon lange Stand der Technik. Neu ist der Einsatz auch auf der Antriebs- oder E/A-Ebene und damit in dem Bereich, der bislang den Feldbussystemen vorbehalten blieb. Hier sind hohe Echtzeitfähigkeit, Eignung auch für kleine Datenmengen und natürlich günstige Kosten gefragt. EtherCAT trägt diesen Forderungen Rechnung und stellt gleichzeitig die Internet Technologien auch auf der E/A Ebene zur Verfügung.

1.2 Ethernet und Echtzeitfähigkeit

Es gibt viele verschiedene Ansätze, mit denen Ethernet echtzeitfähig gemacht werden soll: so wird z.B. das Zugriffsverfahren CSMA/CD durch überlagerte Protokollschichten außer Kraft gesetzt und durch ein Zeitscheibenverfahren oder durch Polling ersetzt; andere Vorschläge sehen spezielle Switches vor, die Ethernet Pakete zeitlich präzise kontrolliert verteilen. Diese Lösungen mögen Datenpakete mehr oder weniger schnell und exakt zu den angeschlossenen Ethernet Knoten transportieren – die Zeiten, die für die Weiterleitung zu den Ausgängen oder Antriebsreglern und für das Einlesen der Eingangsdaten benötigt werden, sind jedoch stark implementierungsabhängig.

Wenn für jeden Teilnehmer individuelle Ethernet Frames Verwendung finden, so ist zudem die Nutzdatenrate prinzipiell sehr gering: Das kürzeste Ethernet Frame ist 84

Bytes lang (incl. Inter Packet Gap IPG). Wenn z.B. ein Antrieb zyklisch 4 Bytes Istwert und Status sendet und entsprechend 4 Bytes Sollwert und Kontrollwort empfängt, so wird bei 100% Buslast (also unendlich kurzer Antwortzeit des Antriebs) nur eine Nutzdatenrate von $4/84= 4,7\%$ erreicht. Bei durchschnittlich $10 \mu\text{s}$ Antwortzeit sinkt die Rate schon auf $1,9\%$. Diese Limitierungen gelten für alle Echtzeit-Ethernet Ansätze, die an jeden Teilnehmer ein Ethernet Frame senden bzw. erwarten – und zwar unabhängig von den verwendeten Protokollen innerhalb des Ethernet Frames.

1.3 Ethernet Funktionsprinzip

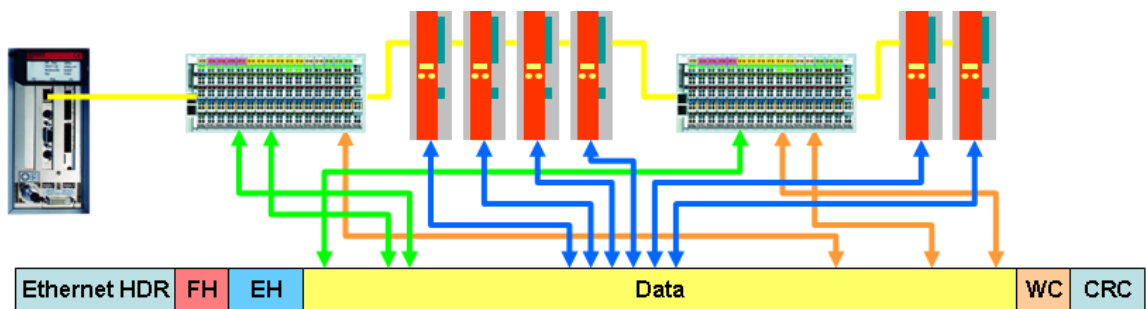


Bild 1: EtherCAT: Prozessdaten werden in Ethernet Telegramm eingefügt

Mit der EtherCAT Technologie werden diese prinzipiellen Begrenzungen anderer Ethernet Lösungen überwunden: das Ethernet Paket wird nicht mehr in jeder Anschaltung zunächst empfangen, dann interpretiert und die Prozessdaten weiterkopiert. Die EtherCAT Slave-Geräte entnehmen die für sie bestimmten Daten, während das Telegramm das Gerät durchläuft.

Ebenso werden Eingangsdaten im Durchlauf in das Telegramm eingefügt (siehe Bild 1). Die Telegramme werden dabei nur wenige Nanosekunden verzögert.

Da ein Ethernet Frame sowohl in Sende- als auch in Empfangsrichtung die Daten vieler Teilnehmer erreicht, steigt die Nutzdatenrate auf über 90% an. Dabei werden die Voll-Duplex Eigenschaften von 100BaseTx vollständig ausgenutzt, so dass effektive Datenraten von $> 100 \text{ Mbit/s}$ ($>90\%$ von $2 \times 100 \text{ Mbit/s}$) erreichbar sind (siehe Bild 2).

Bei 4 Byte Nutzdaten je Knoten:

- Polling / Timeslicing: ca. $2..5\%$
- Broadcast (Master \rightarrow Slaves): ca. $20..30\%$
- Ab 2 Bit Nutzdaten je Knoten:
- **EtherCAT: $80..97\%$**

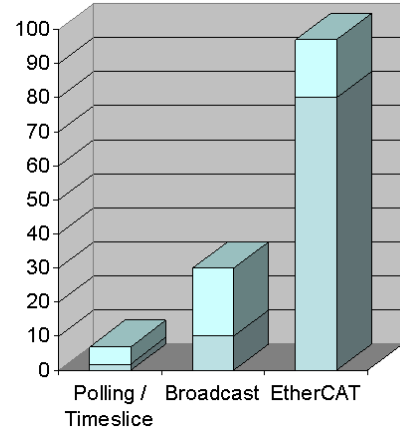


Bild 3: Bandbreiten-Nutzung im Vergleich

1.4 EtherCAT verwendet Standard Ethernet Frames

Das Ethernet Protokoll gemäß IEEE 802.3 bleibt bis in die einzelne Klemme erhalten, der Sub-Bus entfällt. Lediglich die Übertragungsphysik wird im Koppler von Twisted Pair bzw. Lichtleiterphysik auf E-Bus (ein alternativer Ethernet Physical Layer: LVDS) gewandelt, um den Anforderungen der elektronischen Reihenklemme gerecht zu

werden. Die Signalform innerhalb der Klemmenreihe (E-Bus) eignet sich für kurze Strecken – bis 10 m – auch zur Übertragung auf verdrehter Zweidrahtleitung. Damit kann die Klemmenreihe besonders kostengünstig verlängert werden – wobei der anschließende Wechsel auf Ethernet jederzeit möglich ist.

Das für Prozessdaten optimierte EtherCAT Protokoll wird dank eines speziellen Ethertypes direkt im Ethernet-Frame transportiert. Es kann aus mehreren Sub-Telegrammen bestehen, die jeweils einen Speicherbereich des bis zu 4 Gigabyte großen logischen Prozessabbildes bedienen. Die datentechnische Reihenfolge ist dabei unabhängig von der physikalischen Reihenfolge der Ethernet-Klemmen im Netz, es kann wahlfrei adressiert werden. Broadcast, Multicast und Querkommunikation zwischen Slaves sind möglich. Die Übertragung direkt im Ethernet-Frame wird stets dann eingesetzt, wenn EtherCAT-Komponenten im gleichen Subnetz wie der Steuerungsrechner betrieben werden.

Der Einsatzbereich von EtherCAT ist jedoch nicht auf ein Subnetz beschränkt: EtherCAT UDP verpackt das EtherCAT Protokoll in UDP/IP Datagramme (siehe Bild 3). Hiermit kann jede Steuerung mit Ethernet Protokoll-Stack EtherCAT Systeme ansprechen. Selbst die Kommunikation über Router hinweg in andere Subnetze ist möglich. Selbstverständlich hängt die Leistungsfähigkeit des Systems in dieser Variante von den Echtzeiteigenschaften der Steuerung und ihrer Ethernet Protokollimplementierung ab. Die Antwortzeiten des EtherCAT Netzwerks an sich werden jedoch nur minimal eingeschränkt: lediglich in der ersten Station muss das UDP Datagramm entpackt werden.

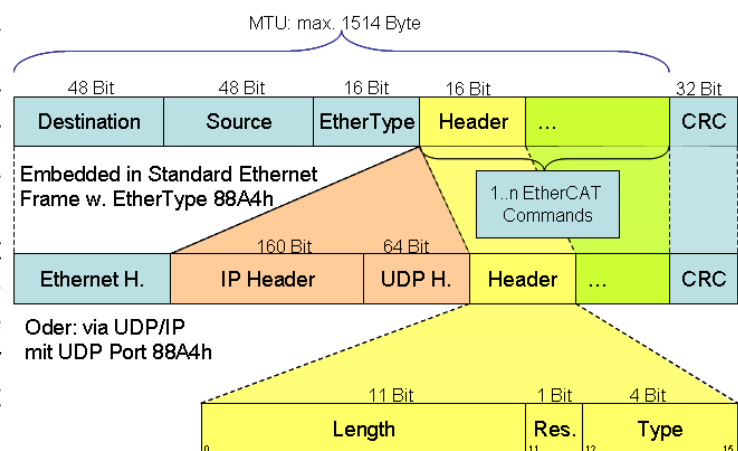


Bild 3: EtherCAT: Standard Frames nach [3]

EtherCAT verwendet ausschließlich Standard-Frames - sie werden nicht verkürzt. Damit können EtherCAT Frames von beliebigen Ethernet Controllern verschickt werden (Master), und Standard Tools (z.B. "Sniffer") können eingesetzt werden.

1.5 Topologie

Linie, Baum oder Stern: EtherCAT unterstützt nahezu beliebige Topologien (siehe Bild 4). Die von den Feldbussen her bekannte Bus- oder Linienstruktur wird damit auch für Ethernet verfügbar.

Besonders praktisch für die Anlagenverdrahtung ist die Kombination aus Linie und Abzweigen bzw. Stichleitungen: die benötigten Schnittstellen sind auf den Kopplern vorhanden, zusätzliche Switches werden nicht benötigt. Natürlich kann aber auch die klassische Switch-basierte Ethernet Sterntopologie eingesetzt werden.

Die maximale Flexibilität bei der Verdrahtung wird durch die Auswahl verschiedener Leitungen vervollständigt. Flexible und sehr preiswerte Standard Ethernet Patch-Kabel übertragen die Signale wahlweise auf Ethernet-Art (100Base-TX) oder in der E-Bus-Signaldarstellung. Kunststoff-Lichtwellenleiter (POF) werden das System für spezielle Anwendungsfälle ergänzen. Die gesamte Bandbreite der Ethernet Vernetzung – wie verschiedenste Lichtleiter und Kupferkabel – kann in der Kombination mit Switches oder Medienumsetzern zum Einsatz kommen.

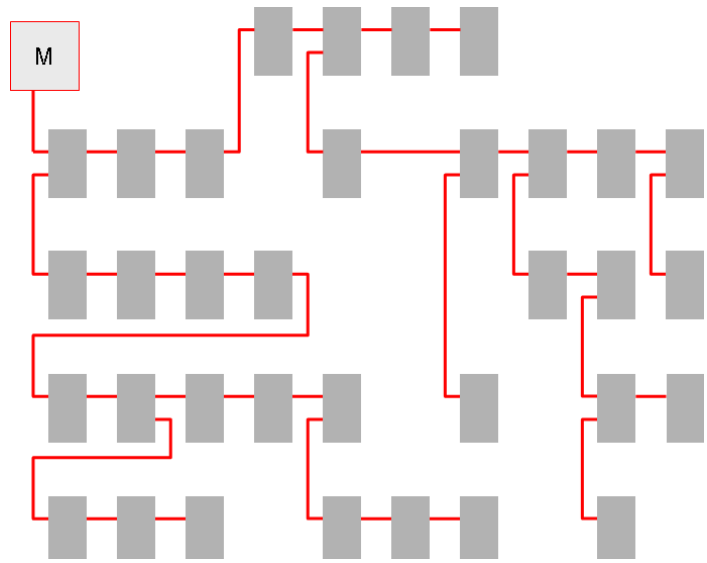


Bild 4: EtherCAT unterstützt flexible Topologien

Die Fast Ethernet Physik erlaubt eine Leitungslänge von 100 m zwischen zwei Teilnehmern, die E-Bus Leitung ist für Abstände bis 10m vorgesehen. Für jede Leitungsstrecke kann die Signalvariante individuell ausgewählt werden. Da bis zu 65535 Teilnehmer angeschlossen werden können, ist die gesamte Netzausdehnung nahezu unbeschränkt.

1.6 Performance

Mit EtherCAT werden neue Dimensionen in der Netzwerk-Performance erreicht. Dank ASIC-Integration im Slave und DMA Zugriff auf die Netzwerkkarte im Master erfolgt die gesamte Protokollbearbeitung in Hardware und ist damit unabhängig von der Laufzeit von Protokollstacks, von CPU Performance oder Software-Implementierung. Die Update Zeit für 1000 E/As beträgt nur 30 μ s – einschließlich I/O-Durchlaufzeit. Mit einem einzigen Ethernet-Frame können bis zu 1486 Bytes Prozessdaten ausgetauscht werden – das entspricht fast 12000 digitalen Ein- und Ausgängen. Für die Übertragung dieser Datenmenge werden dabei nur 300 μ s benötigt.

Für die Kommunikation mit 100 Servoachsen werden nur 100 μ s benötigt. In dieser Zeit werden alle Achsen mit Sollwerten und Steuerdaten versehen und melden ihre Ist-Position und Status. Durch das Distributed Clock Verfahren können die Achsen dabei mit einer Abweichung von deutlich weniger als einer Mikrosekunde synchronisiert werden.

Die extrem hohe Performance der EtherCAT Technologie ermöglicht Steuerungs- und Regelungskonzepte, die mit klassischen Feldbussystemen nicht realisierbar waren. Mit EtherCAT steht eine Kommunikationstechnologie zur Verfügung, die der überlegenen Rechenleistung moderner Industrie-PCs entspricht. Das Bussystem ist nicht mehr der Flaschenhals im Steuerungskonzept. Verteilte E/As werden schneller erfasst, als dies mit den meisten lokalen E/A-Schnittstellen möglich ist. Das Ether-

CAT Technologieprinzip ist skalierbar und nicht an die Baudrate von 100 MBaud gebunden – eine Erweiterung auf GBit Ethernet ist möglich.

1.7 Diagnose

Die Erfahrungen mit Feldbussystemen zeigen, dass die Verfügbarkeit und Inbetriebnahmezeiten entscheidend von der Diagnosefähigkeit abhängen. Nur eine schnell und präzise erkannte und eindeutig lokalisierbare Störung kann kurzfristig behoben werden. Deshalb wurde bei der Entwicklung des EtherCAT Systems besonderer Wert auf vorbildliche Diagnoseeigenschaften gelegt.

Bei der Inbetriebnahme gilt es zu prüfen, ob die Ist-Konfiguration mit der Soll-Konfiguration übereinstimmt. Auch die Topologie sollte der Konfiguration entsprechen. Durch die eingebaute Topologie-Erkennung kann nicht nur diese Überprüfung beim Systemstart stattfinden – auch ein automatisches Einlesen des Netzwerkes ist möglich (Konfigurations-Upload).

Bitfehler in der Übertragung werden durch die Auswertung der CRC-Prüfsumme zuverlässig erkannt: das 32 Bit CRC-Polynom weist eine minimale Hamming-Distanz von 4 auf. Neben der Bruchstellenerkennung und -lokalisierung erlauben Protokoll, Übertragungsphysik und Topologie des EtherCAT Systems eine individuelle Qualitätsüberwachung jeder einzelnen Übertragungsstrecke. Die automatische Auswertung der entsprechenden Fehlerzähler ermöglicht die exakte Lokalisierung kritischer Netzwerkabschnitte. Schleichende oder wechselnde Fehlerquellen wie EMV-Einflüsse, fehlerhafte Steckverbindungen oder Kabelschäden werden erkannt und lokalisiert, auch wenn sie die Selbstheilungsfähigkeit des Netzwerkes noch nicht überfordern.

2 Masterimplementierung: Software-Tools und Aufwand

EtherCAT ist die einzige Ethernet Lösung für harte Echtzeit-Anforderungen, die in der Master-Anschaltung ohne Kommunikations-Coprozessor auskommt. Damit ist keine dedizierte Einsteckkarte erforderlich, ein Standard Ethernet Controller genügt. Der Master ist damit in der Regel eine reine Softwarelösung.

2.1 EtherCAT/UDP

Für Applikationen mit geringeren Echtzeitanforderungen und bei denen die Routing Funktionalität von IP gefragt ist, empfiehlt sich die UDP/IP Protokollvariante von EtherCAT. Hier kann die Implementierung auf dem Socket-Interface eines beliebigen TCP/IP Stacks aufsetzen. Diese Lösung ist z.B. für die Gebäudeautomation interessant: hier spielt die Echtzeitfähigkeit von EtherCAT eine untergeordnete Rolle. Im Vordergrund steht hier die einfache und flexible Verdrahtung. So können z.B. Raumcontroller oder E/A Schnittstellen (wie etwa zur Licht- und Sonnenschutz-Ansteuerung) in beliebiger Anzahl entlang der Außenfassade "in Reihe" verbunden werden, ohne dass sternförmig auf Switches verkabelt werden muss.

2.2 EtherCAT Master für einfache Steuerungen

Auch die Implementierung eines EtherCAT Masters für die kleine und mittlere Steuerungstechnik ist sehr einfach. Betrachtet wird eine SPS mit einem einzigen Prozessabbild: wenn dieses 1488 Bytes nicht übersteigt, so genügt das zyklische Versenden eines einzigen Ethernet Frames – und zwar mit der Zykluszeit der SPS. Da sich der Header zur Laufzeit nicht ändert, muss also lediglich ein konstanter "Vorspann" zum Prozessabbild hinzugefügt und das Ergebnis dem Ethernet Controller übergeben werden (siehe Bild 5).

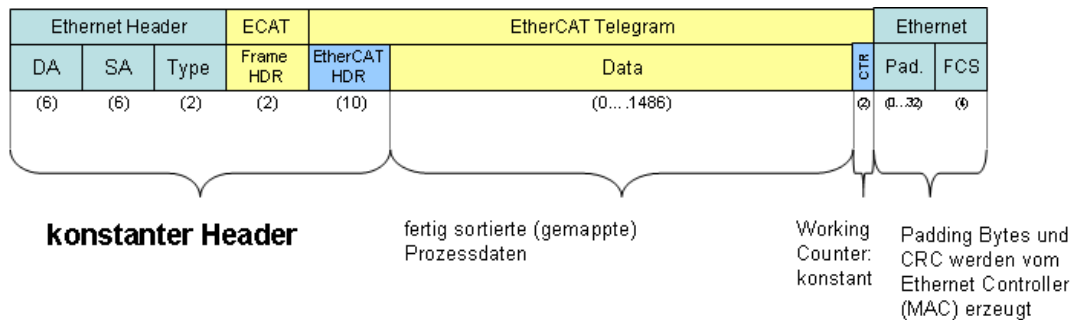


Bild 5: Masterimplementierung mit einem Prozessabbild

Dabei ist das Prozessabbild bereits fertig sortiert, da das Mapping bei EtherCAT ja nicht im Master, sondern in den Slaves erfolgt – die Peripheriegeräte fügen Ihre Daten ja an die entsprechende Stelle im durchlaufenden Frame ein.

2.3 EtherCAT Master für komplexe Steuerungen

Für komplexe Automatisierungsaufgaben bietet EtherCAT noch vielfältige zusätzliche Features: die Protokollstruktur erlaubt das individuelle Update verschiedener Prozessabbilder mit unterschiedlichen Zykluszeiten, hart echtzeitfähigen Querverkehr zwischen Slave-Geräten und Synchronisierung mit verteilten Uhren. Auch das Durchleiten von allgemeinen Ethernet Frames ist ein wichtiges Feature, das vom Master unterstützt werden kann.

2.4 Konfiguration

Einen hohen Anteil am Implementierungsaufwand von Feldbus-Mastern nimmt in der Regel das Konfigurationstool ein. Um diesen zu minimieren, wurden klar definierte Schnittstellen geschaffen: Per XML File erhält das Konfigurationstool Kenntnis von den Geräteeigenschaften (ggf. können diese auch direkt aus dem Gerät hochgeladen werden). Das Konfigurationstool erzeugt seinerseits auch ein XML File, das alle relevanten Informationen über Netzwerkstruktur, Prozessdatenlayout, Aufstarten und Diagnose enthält. Die eigentliche Masterimplementierung muss lediglich dieses File einlesen und die benötigten Ethernet Frames für Start und Betrieb des Netzes extrahieren.

Stand-Alone Konfigurationstools, die diesen Schnittstellenvereinbarungen entsprechen, sind von mehreren Herstellern angekündigt. Diese können in Verbindung mit

beliebigen Master-Implementierungen verwendet werden. Auch TwinCAT, die Automationssoftware von Beckhoff, unterstützt diese Schnittstellen.

2.5 Master Sample Code

Zur Unterstützung einer Master-Implementierung steht ein Master-Sample Code gegen eine Schutzgebühr zur Verfügung. Die Software wird im Source-Code ausgeliefert und umfasst alle Funktionen für einen EtherCAT Master samt Ethernet over EtherCAT. Der Code muss lediglich noch an die Zielhardware und das verwendete RTOS angepasst werden – er wurde für Windows Umgebungen erstellt (siehe Bild 6).

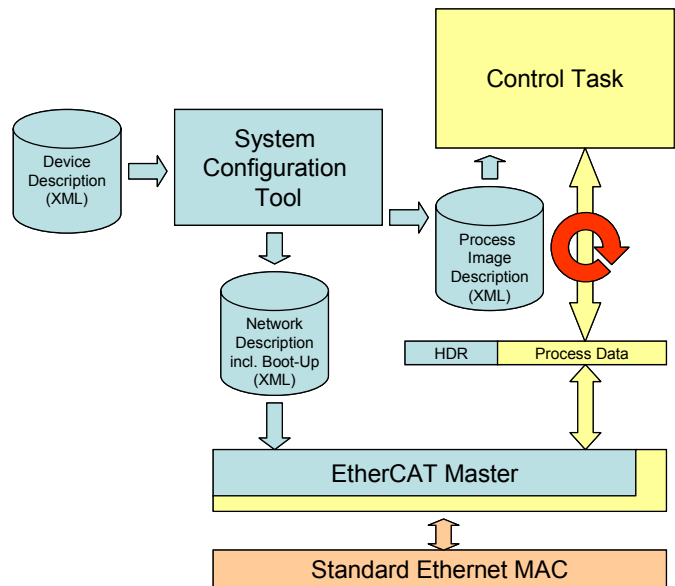


Bild 6: EtherCAT Master Sample Code

2.6 Monitor Tools

Da EtherCAT Standard Ethernet Frames nach IEEE 802.3 verwendet eignet sich jedes handelsübliche Ethernet Monitor Tool zur Beobachtung der EtherCAT Kommunikation. Zusätzlich gibt es kostenlose Parser-Software für Ethereal (ein open source Monitor Tool) und den Microsoft Netzwerk Monitor, mit der mitgeschnittener EtherCAT Datenverkehr komfortabel aufbereitet und zur Anzeige gebracht wird.

3 Slave Geräte: Schnittstellen und Technologiebausteine

3.1 Slave Controller-Schnittstellen

Um sowohl ganz einfache Teilnehmer wie z.B. ein 2-Bit digitales Eingangsmodul, als auch komplexe Teilnehmer wie ein Feldbusmastermodul, gleichermaßen kosten- und performance-optimal zu implementieren, verfügt der EtherCAT Slave Controller über eine Auswahl an Schnittstellen.

3.1.1 SSI-Slave

Das SSI-Slave-Interface ist für alle Geräte mit kleiner Prozessdatenmenge geeignet, wie z.B. analoge Module, Geber, Encoder oder auch einfache Antriebe. Der Slave Controller verfügt über einen 4 kByte großen Speicher (DPRAM), der über EtherCAT adressiert werden kann. Die Datenübertragung vom Slave-Controller zum Microcontroller erfolgt über ein bis zu 10 MBit schnelles SSI-Interface, das vom Buszugriff entkoppelt ist. Die SSI-Übertragung stellt eine kostengünstige Variante für Geräte mit wenigen Prozessdatenbytes dar. Die Menge der azyklischen Parameter- und Diagnosedaten ist nahezu unbeschränkt, sofern deren Übertragung nicht zeitkritisch ist.

3.1.2 32-Bit-Parallel-In/Out

Das 32-Bit-Parallel-In/Out-Interface eignet sich für den Anschluss von bis zu 32 digitalen Ein- bzw. Ausgängen, aber auch für einfache Aktoren oder Sensoren, die mit 32 Datenbits auskommen. Im Gegensatz zum SSI-Slave-Interface ist der Zugriff auf die Prozessdaten wesentlich schneller, und es ist auch kein Microcontroller erforderlich. Allerdings ist das Datenaufkommen auch auf die 32 Bits beschränkt.

Das Interface lässt sich als

- 32 In
- 32 Out
- 16 In / 16Out
- 24 In / 8 Out
- 8 Out / 24 In

konfigurieren.

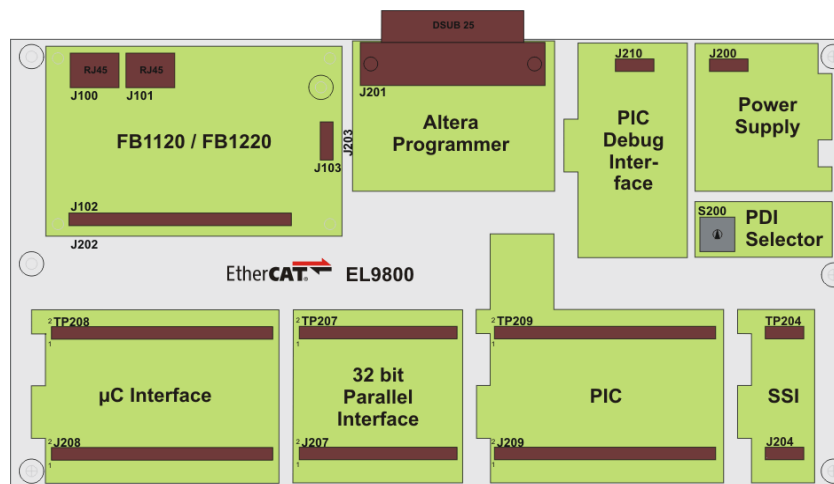


Bild 7: EtherCAT Slave Evaluation Kit

3.1.3 Paralleles 8/16-Bit-Microcontroller-Interface

Das parallele 8/16-Bit-Microcontroller-Interface entspricht herkömmlichen Schnittstellen bei Feldbus-Controllern mit DPRAM-Schnittstelle. Diese Anschaltung ist etwas aufwändiger aber dafür auch leistungsfähiger als die anderen Schnittstellen und eignet sich daher insbesondere für komplexe Teilnehmer mit größerem Datenaufkommen wie z.B. Antriebsregler.

3.2 Slave Sample Software

Zur Bedienung der Microcontroller Schnittstelle und des SSI Interface gehören Beispielprogramme (C Source Code) zum Lieferumfang des EtherCAT Evaluation Kits.

3.3 Evaluation Kit

Das Evaluation Kit (siehe Bild 7) bietet die Möglichkeit, mit den verschiedenen Schnittstellen (Process Data Interfaces – PDI) des EtherCAT Slave Controllers zu

arbeiten. Dank der aufsteckbaren EtherCAT Anschaltung kann wahlweise die 100BaseTx- als auch die E-Bus-Physik gewählt werden. Die einzelnen PDI's sind per Hardware und Software umschaltbar. Optional wird ein Workshop zum Evaluation Kit angeboten, in dem mit allen PDI's gearbeitet wird und die Grundlagen der Hardware- und Softwareimplementierung behandelt werden.

4 Geräteprofilintegration und Nutzung vorhandener Software

Die Geräteprofile bilden die Schnittstelle, die der Anwender vom Feldbussystem hauptsächlich sieht – die eigentlichen Kommunikationsdienste sind meist versteckt und stellen Prozess-, Parameter- und Diagnosedaten mit unterschiedlichen zeitlichen Randbedingungen zur Verfügung. Die Geräteprofile geben dabei vor, wie die Geräteparameter abgelegt sind, und definieren das Geräteverhalten. Es hat viele Jahre gedauert, die Feldbus-Geräteprofile zu entwickeln und im Markt zu etablieren.

Für EtherCAT werden deshalb nur dort eigene Geräteprofile entwickelt, wo keine geeigneten vorhanden sind. In den anderen Bereichen nutzt EtherCAT bewährte und eingeführte Profile – wie sie z.B. für CANopen, Profibus oder Sercos existieren. Diese Profildfamilien sind sehr umfangreich und reichen von E/A Geräten und Encodern über Antriebe bis hin zu Hydraulikventilen. Der CANopen Ansatz geht dabei auch über die reine Gerätesicht hinaus: in Applikationsprofilen sind die Kommunikationsschnittstellen ganzer mechatronischer Subsysteme spezifiziert.

EtherCAT bildet die Kommunikationsdienste so passend ab, dass selbst Teile einer ggf. vorhandenen Protokollimplementierung wieder verwendet werden können, beginnend zunächst mit CANopen. Für Neuentwicklungen stehen zudem Funktionserweiterungen zur Verfügung, die der größeren Bandbreite von EtherCAT Rechnung tragen.

EtherCAT ist dabei nicht auf die CANopen Profile beschränkt, es können in Zukunft weitere Profile unterstützt werden. Bei Bedarf können auch entsprechende Schnittstellen zu anderen Standards bereitgestellt werden.

5 Zusammenfassung und Ausblick

EtherCAT ist nicht nur eines der leistungsfähigsten Industrial Ethernetsysteme, es ist aufgrund der verschiedenen Prozeßdateninterfaces des Slave Controllers auch flexibel und leicht zu implementieren sowohl in einfachen wie in komplexen Geräten. Die zur Verfügung stehende Slave Sample Software, das Evaluation Kit, Workshops und bei komplexeren Geräten die Möglichkeit vorhandene Profile zu verwenden bieten hier umfassende Unterstützung und machen die Entwicklung von EtherCAT Slaves so einfach wie möglich.

Auch die Entwicklung von EtherCAT Mastern ist durch den Master Sample Code, XML-Schnittstelle und den implementierbaren Konfigurator so offen und einfach wie möglich gestaltet.