

به نام خدا

مستند پروتکل EtherCAT

استاد فصحتی

درس مدارهای واسط

محمدفرحان بهرامی

۴۰۱۱۰۵۷۲۹

۵ کاربرد پروتکل ETHERCAT

- ۵..... ۲. سیستم‌های کنترل عددی (CNC)
- ۵..... ۳. سیستم‌های تست و اندازه‌گیری (Test & Measurement Systems)
- ۵..... ۴. اتوماسیون صنایع خودرو (Automotive Industry)
- ۵..... ۵. نیروگاه‌ها و سیستم‌های انرژی (Power & Energy Systems)
- ۵..... ۶. صنایع بسته‌بندی و چاپ (Packaging & Printing)
- ۵..... ۷. حمل‌ونقل ریلی و سیستم‌های کنترل ترافیک (Rail & Traffic Control)
- ۵..... ۸. سیستم‌های پزشکی (Medical Systems)
- ۵..... ۹. اینترنت اشیا صنعتی (IIoT)

۶ علت توسعه این پروتکل ETHERCAT

- ۷..... ۱. نیاز به سرعت بالا و تأخیر کم در اتوماسیون صنعتی
- ۷..... ۲. افزایش پیچیدگی و تعداد دستگاه‌های متصل در سیستم‌های صنعتی
- ۷..... ۳. نیاز به دقت بالا در هماهنگ‌سازی سیستم‌های کنترلی (Synchronization)
- ۷..... ۴. کاهش هزینه‌های سخت‌افزاری و بهینه‌سازی مصرف منابع
- ۷..... ۵. سازگاری با اینترنت استاندارد و نیاز به یکپارچگی سیستم‌ها
- ۷..... ۶. نیاز به افزایش امنیت و کاهش خطاهای ارتباطی

۷ اتصالات و مدارات لایه فیزیکی این پروتکل ETHERCAT

- ۷..... توضیحات مربوط به شکل ۱:
- ۸..... انواع اتصالات در پروتکل EtherCAT

۸ نوع ارتباط پروتکل

۸ نوع انکودینگ

۸ نحوه تولید سیگنال

۹ روش انتقال سیگنال

۹ قابلیت اتصال چندین دستگاه/ماژول سخت‌افزاری در این پروتکل

۹ نحوه اتصال دستگاه‌ها

۹ مدیریت برخورد در این پروتکل

۹ کنترل متمرکز توسط Master

۹ زمان‌بندی دقیق و حلقه ارتباطی Real-Time

۹ ساختار تعیین موقعیت داده‌ها در فریم:

۱۰	آدرس دهی در این پروتکل
۱۰	آدرس دهی خودکار (Auto-Increment Addressing):
۱۰	آدرس دهی ثابت (Configured Addressing):
۱۰	آدرس دهی مبتنی بر اترنت استاندارد (Ethernet MAC Addressing):
۱۰	مسیریابی در این پروتکل
۱۰	ساختار زنجیره‌ای و پردازش مستقیم فریم‌ها:
۱۰	مکانیزم Loopback در صورت نیاز:
۱۰	چرا EtherCAT نیازی به مسیریابی پیچیده ندارد؟
۱۱	توضیحات مربوط به شکل ۳
۱۲	نحوه مدیریت جریان داده در این پروتکل
۱۲	۱. تنظیمات اولیه و پیکربندی شبکه
۱۲	۲. پردازش اطلاعات و ارسال به Master
۱۳	۳. اجرای دستورات و پردازش داده‌ها
۱۳	۴. اجرای وظایف کنترلی و تبادل داده
۱۳	۵. ارسال داده به شبکه و ارتباط با Slave‌ها
۱۳	مدیریت و تشخیص خطا در لایه‌های مختلف این پروتکل
۱۳	لایه فیزیکی (Physical Layer)
۱۳	مثال
۱۳	لایه پیوند داده (Data Link Layer)
۱۴	مثال
۱۴	لایه شبکه (Network Layer)
۱۴	مثال
۱۴	لایه انتقال (Transport Layer)
۱۴	مثال
۱۴	لایه کاربردی (Application Layer)
۱۵	مثال
۱۵	انواع پیام در این پروتکل
۱۵	پیام‌های فرآیند داده‌ها (Process Data Messages - PDM)
۱۵	پیام‌های پارامتری و تنظیمات (Service Data Messages - SDM)
۱۵	پیام‌های دسترسی به حافظه (Mailbox Messages)

۱۵	فرمت کلی پیام ها در این پروتکل
۱۵	توضیحات مربوط به شکل ۵:
۱۵	Ethernet Header (Ethernet H.)
۱۶	EtherCAT Data
۱۶	Header
۱۶	EtherCAT Datagrams
۱۶	Datagram Header (Datagram H.)
۱۶	Data
۱۶	WKC (Working Counter)
۱۶	FCS (Frame Check Sequence)
۱۶	رویکرد تشخیص خطا
۱۶	ساختار مبتنی بر Master/Slave
۱۶	استفاده از مکانیسم CRC در لایه Data Link
۱۶	شماره گذاری بسته ها با Working Counter (WKC)
۱۶	رویکرد تصحیح خطا
۱۷	بازارسال خودکار پیام توسط Master
۱۷	استفاده از Redundancy برای جلوگیری از خطا
۱۷	بررسی و تصحیح خطای بیت (Bit Error Handling)
۱۷	منابع

کاربرد پروتکل EtherCAT

EtherCAT (Ethernet for Control Automation Technology) یک پروتکل اتترنت صنعتی است که برای کاربردهای کنترل

بلادرنگ طراحی شده است. که کاربرد های آن به صورت زیر است:

۱. کنترل رباتیک (Robotics Control)

EtherCAT به دلیل زمان پاسخ بسیار پایین (کمتر از ۱۰۰ میکروثانیه)، برای کنترل همزمان چندین محور حرکتی در ربات‌ها ایده‌آل است. این پروتکل در ربات‌های صنعتی، بازوهای رباتیک، و AGV (وسایل نقلیه هدایت‌شونده خودکار) استفاده می‌شود.

۲. سیستم‌های کنترل عددی (CNC)

در ماشین‌های CNC و دستگاه‌های چاپ سه‌بعدی، EtherCAT برای هماهنگ‌سازی دقیق حرکات موتورهای پله‌ای (Stepper Motors) و سروو موتورها استفاده می‌شود.

۳. سیستم‌های تست و اندازه‌گیری (Test & Measurement Systems)

در تجهیزات آزمایشگاهی و تست خودکار، EtherCAT داده‌ها را با سرعت بالا منتقل می‌کند، که باعث افزایش دقت در جمع‌آوری داده‌ها می‌شود. در دستگاه‌های اندازه‌گیری ارتعاشات، دما، فشار، و سایر پارامترهای صنعتی به کار می‌رود.

۴. اتوماسیون صنایع خودرو (Automotive Industry)

در خطوط تولید خودروسازی برای کنترل هماهنگ ربات‌ها، مونتاژ قطعات، و تست کیفیت استفاده می‌شود. در سیستم‌های HIL (Hardware-in-the-Loop) برای تست و شبیه‌سازی کنترل خودروهای الکتریکی و خودران کاربرد دارد.

۵. نیروگاه‌ها و سیستم‌های انرژی (Power & Energy Systems)

در نیروگاه‌های خورشیدی، بادی، و شبکه‌های هوشمند (Smart Grid) برای کنترل و پایش توزیع برق استفاده می‌شود. در سیستم‌های ذخیره انرژی و کنترل باتری برای بهینه‌سازی توزیع انرژی کاربرد دارد.

۶. صنایع بسته‌بندی و چاپ (Packaging & Printing)

در دستگاه‌های بسته‌بندی دقیق و پرسرعت، برای هماهنگ‌سازی چندین موتور و سنسور کاربرد دارد. در صنعت چاپ دیجیتال، برای کنترل جوهرپاش‌ها و هدای چاپی استفاده می‌شود.

۷. حمل‌ونقل ریلی و سیستم‌های کنترل ترافیک (Rail & Traffic Control)

در قطارهای پرسرعت و مترو، برای هماهنگ‌سازی عملکرد تجهیزات مختلف مانند ترمزها، درب‌ها، و نمایشگرهای اطلاعاتی استفاده می‌شود. در سیستم‌های کنترل ترافیک هوشمند (ITS) برای مدیریت چراغ‌های راهنمایی، دوربین‌ها، و نظارت بر جاده‌ها به کار می‌رود.

۸. سیستم‌های پزشکی (Medical Systems)

در دستگاه‌های تصویربرداری پزشکی مانند MRI و CT-Scan برای انتقال داده‌های سنسورهای دقیق استفاده می‌شود. در ربات‌های جراحی که نیاز به تأخیر کم و دقت بالا دارند، EtherCAT به عنوان پروتکل ارتباطی به کار می‌رود.

۹. اینترنت اشیا صنعتی (IIoT)

EtherCAT در کارخانه‌های هوشمند برای ارتباط سریع بین سنسورها، کنترلرها، و سیستم‌های ابری جهت پایش بلادرنگ و تعمیر و نگهداری پیش‌بینی‌شده (Predictive Maintenance) استفاده می‌شود.

علت توسعه این پروتکل EtherCAT

پروتکل EtherCAT (Ethernet for Control Automation Technology) توسط شرکت Beckhoff Automation در اوایل دهه ۲۰۰۰ توسعه یافت. دلایل اصلی توسعه این پروتکل شامل موارد زیر است:

۱. نیاز به سرعت بالا و تأخیر کم در اتوماسیون صنعتی
پروتکل‌های قدیمی مانند Modbus، PROFIBUS و CANopen دارای تأخیر بالا و سرعت پایین بودند که باعث مشکلاتی در هماهنگی دستگاه‌های صنعتی می‌شد. EtherCAT با استفاده از پردازش در حال عبور (Processing on the Fly)، داده‌ها را بدون نیاز به ذخیره‌سازی در هر نود پردازش می‌کند، که باعث کاهش شدید تأخیر (کمتر از ۱۰۰ میکروثانیه) و افزایش سرعت انتقال داده‌ها (تا ۱۰۰ مگابیت بر ثانیه) می‌شود.
۲. افزایش پیچیدگی و تعداد دستگاه‌های متصل در سیستم‌های صنعتی
در کارخانه‌های مدرن، تعداد سنسورها، عملگرها و ربات‌های متصل افزایش یافته است، که پروتکل‌های سنتی پاسخگوی این حجم از داده نبودند. EtherCAT قابلیت پشتیبانی از حداکثر ۶۵,۵۳۵ دستگاه را دارد و بدون نیاز به سخت‌افزار اضافی، ارتباطات همزمان را ممکن می‌سازد.
۳. نیاز به دقت بالا در هماهنگ‌سازی سیستم‌های کنترلی (Synchronization)
بسیاری از کاربردهای صنعتی مانند کنترل رباتیک، CNC، و خطوط تولید خودکار نیاز به زمان‌بندی دقیق و هماهنگی در سطح میکروثانیه دارند. EtherCAT دارای مکانیزم توزیع زمان دقیق (Distributed Clocks) است که زمان‌بندی نودهای مختلف را با دقت نانو ثانیه هماهنگ می‌کند.
۴. کاهش هزینه‌های سخت‌افزاری و بهینه‌سازی مصرف منابع
در بسیاری از پروتکل‌های قدیمی، برای اتصال تعداد زیادی دستگاه نیاز به سوئیچ‌های اضافی یا سخت‌افزارهای گران‌قیمت بود. EtherCAT از ساختار زنجیره‌ای (Daisy Chain) و توپولوژی انعطاف‌پذیر (خطی، حلقه‌ای، درختی) پشتیبانی می‌کند، که نیاز به سوئیچ‌های گران‌قیمت را از بین می‌برد و هزینه‌های پیاده‌سازی را کاهش می‌دهد.
۵. سازگاری با اترنت استاندارد و نیاز به یکپارچگی سیستم‌ها
بسیاری از صنایع نیاز به یک پروتکل سریع، سازگار با اترنت استاندارد و بدون نیاز به تغییر سخت‌افزار داشتند. EtherCAT از کابل‌های استاندارد Ethernet و بدون نیاز به تراشه‌های خاص استفاده می‌کند و با TCP/IP، UDP، و سایر پروتکل‌های شبکه سازگار است.
۶. نیاز به افزایش امنیت و کاهش خطاهای ارتباطی
در محیط‌های صنعتی، نویزهای الکترومغناطیسی می‌توانند باعث خطا در ارسال داده‌ها شوند. EtherCAT دارای مکانیزم‌های تشخیص و اصلاح خطا (Error Detection & Redundancy) است و از توپولوژی حلقه‌ای (Ring Topology) برای افزایش تحمل‌پذیری در برابر خرابی‌ها پشتیبانی می‌کند.

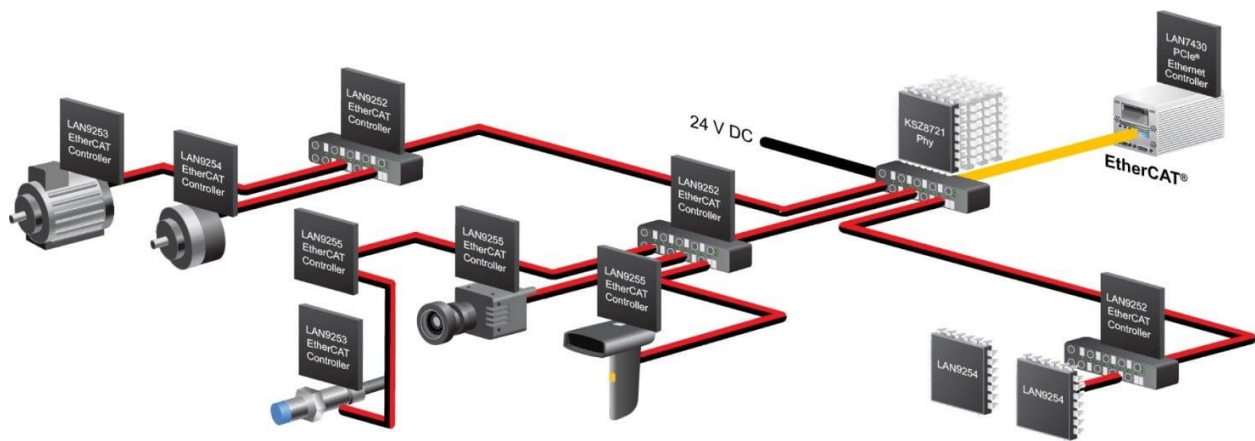
اتصالات و مدارات لایه فیزیکی این پروتکل EtherCAT

EtherCAT از استاندارد IEEE ۸۰۲٫۳ تبعیت می‌کند و در سطح سخت‌افزاری از سیگنالینگ تفاضلی (Differential Signaling) برای کاهش نویز و افزایش کیفیت ارتباط استفاده می‌کند. در BASE-TX۱۰۰، دو زوج سیم به کار می‌روند:

❖ TX+ / TX- (ارسال داده)

❖ RX+ / RX- (دریافت داده)

در سیگنالینگ تفاضلی، داده‌ها از طریق دو سیم با پتانسیل الکتریکی معکوس ارسال می‌شوند، که باعث می‌شود نویزهای الکتریکی تأثیر کمتری روی سیگنال داشته باشند.



شکل (۱) معماری شبکه EtherCAT در یک سیستم صنعتی

توضیحات مربوط به شکل ۱:

- ❖ در سمت راست، بالای تصویر، یک کنترلر اصلی مبتنی بر (LAN۷۴۳۰) PCIe Ethernet Controller وجود دارد که با کابل زردرنگ به شبکه EtherCAT متصل است.
- ❖ تراشه KSZ۸۷۲۱ Phy در مسیر ارتباطی دیده می‌شود که وظیفه تبدیل سیگنال‌های دیجیتال به فرمت قابل انتقال در کابل اترنت را بر عهده دارد.
- ❖ کنترلرهای EtherCAT مانند LAN۹۲۵۲، LAN۹۲۵۳، LAN۹۲۵۴ و LAN۹۲۵۵ در مسیر ارتباطی مشاهده می‌شوند. این تراشه‌ها معمولاً به سنسورها، موتورها، دوربین‌ها و سایر دستگاه‌های صنعتی متصل هستند.
- ❖ کابل‌های قرمز رنگ نشان‌دهنده اتصال دیتا هستند، در حالی که کابل مشکی نشان‌دهنده تغذیه ۲۴ ولت DC برای دستگاه‌های مختلف است.
- ❖ در این معماری، دستگاه‌ها به صورت سری (Daisy Chain) متصل شده‌اند که یکی از مزایای EtherCAT محسوب می‌شود، زیرا نیازی به سوئیچ اضافی ندارد و تأخیر انتقال داده کاهش می‌یابد.

ضروری:

- ❖ Ethernet (RJ45 یا صنعتی M12/M18) برای انتقال داده‌ها.
- ❖ تغذیه برق (V DC24) برای تأمین انرژی دستگاه‌های متصل.
- ❖ سیگنال‌های تفاضلی TX+ / TX- و RX+ / RX- برای انتقال داده‌ها.

اختیاری:

- ❖ ورودی/خروجی دیجیتال و آنالوگ (در برخی کنترلرهای EtherCAT مانند LAN۹۲۵۲ پورت‌های I/O وجود دارند).
- ❖ کابل فیبر نوری (در برخی کاربردهای خاص برای افزایش برد ارتباطی).
- ❖ پورت‌های اضافی برای Redundancy (جهت افزایش قابلیت اطمینان در برخی توپولوژی‌های صنعتی).

نوع ارتباط پروتکل

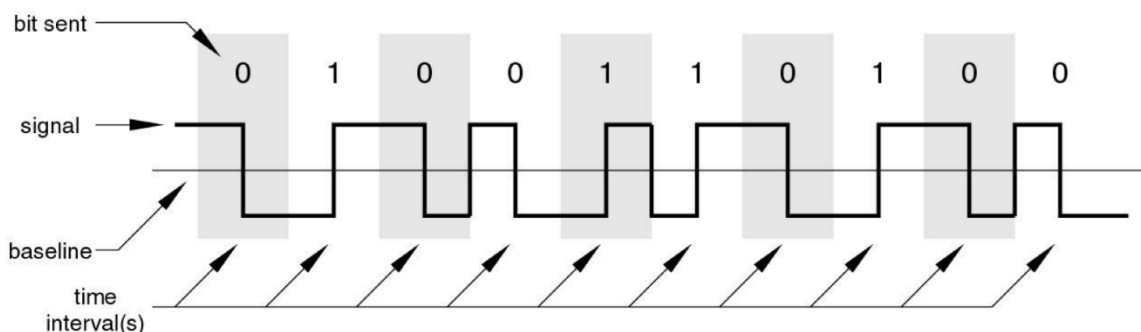
EtherCAT یک پروتکل سریال است. این پروتکل داده‌ها را در یک ساختار حلقه‌ای (Ring) یا زنجیره‌ای (Daisy Chain) از طریق کابل‌های اترنت انتقال می‌دهد. برخلاف پروتکل‌های موازی که چندین خط داده برای انتقال هم‌زمان دارند.

نوع انکودینگ

EtherCAT از انکودینگ منچستر (Manchester Encoding) استفاده می‌کند که در استاندارد BASE-TX۱۰۰ اترنت تعریف شده است. در این روش، هر بیت داده با یک انتقال سطح ولتاژ در وسط دوره زمانی بیت نمایش داده می‌شود که باعث می‌شود کلاک و داده هم‌زمان ارسال شوند و نیاز به سیگنال کلاک جداگانه نباشد. انکودینگ منچستر مزایایی مانند کاهش نویز و تشخیص آسان خطاها را دارد.

نحوه تولید سیگنال

Manchester Encoding



شکل ۲) این پروتکل از انکودینگ منچستر استفاده می‌کند و به همین دلیل در سیگنال دهی آن وقتی در هر دوره زمانی از ارسال سیگنال، سطح سیگنال از ۰ به ۱ گذر کند به معنای بیت ۱ است و اگر از ۱ به ۰ گذر کند به معنای بیت ۰ می‌باشد.

روش انتقال سیگنال

EtherCAT یک پروتکل هم‌زمان (Synchronous) است که به صورت کنترل شده با زمان (Real-Time) کار می‌کند. در این روش، تمام دستگاه‌های متصل به شبکه EtherCAT بر اساس یک کلاک مرجع (مرکز کنترل) داده‌ها را ارسال و دریافت می‌کنند.

قابلیت اتصال چندین دستگاه/ماژول سخت‌افزاری در این پروتکل

پروتکل EtherCAT (Ethernet for Control Automation Technology) به طور خاص برای اتصال چندین دستگاه/ماژول سخت‌افزاری طراحی شده است. این پروتکل یک گذرگاه صنعتی مبتنی بر اترنت است که امکان ارتباط با تعداد زیادی از دستگاه‌ها را در یک شبکه فراهم می‌کند.

نحوه اتصال دستگاه‌ها

دستگاه‌ها در یک شبکه خطی (Daisy Chain) یا حلقه‌ای (Ring Topology) متصل می‌شوند. کنترلر (Master) یک فریم داده‌ی واحد را به همه‌ی دستگاه‌ها ارسال می‌کند. این فریم هنگام عبور از هر دستگاه، داده‌های مربوط به آن دستگاه را دریافت و ارسال می‌کند. از آنجایی که فریم از تمامی دستگاه‌ها عبور می‌کند، نیازی به ارسال پیام‌های جداگانه به هر دستگاه نیست، که باعث کاهش تأخیر و افزایش کارایی می‌شود.

همچنین EtherCAT محدود به دو دستگاه نیست چون برخلاف برخی گذرگاه‌های سنتی مانند SPI یا I²C که محدود به یک ارتباط نقطه‌به‌نقطه یا چند برده با مدیریت برخورد پیچیده هستند، به گونه‌ای طراحی شده که می‌تواند تعداد زیادی دستگاه را در یک شبکه پشتیبانی کند. این ویژگی به دلیل طراحی خاص در مدیریت فریم داده‌ها، زمان‌بندی ارتباط و استفاده از یک Master واحد امکان‌پذیر شده است.

مدیریت برخورد در این پروتکل

EtherCAT برخلاف اترنت استاندارد (Ethernet TCP/IP) که از روش‌هایی مثل CSMA/CD (Carrier Sense Multiple Access with Collision Detection) برای مدیریت برخوردها استفاده می‌کند، نیازی به این مکانیسم‌ها ندارد. دلایل این موضوع عبارت‌اند از:

کنترل متمرکز توسط Master:

فقط یک دستگاه (Master) ارسال‌کننده فریم است، و دستگاه‌های Slave تنها داده‌های خود را درون این فریم تغییر می‌دهند. این ساختار باعث حذف برخوردهای داده‌ای (Data Collision) می‌شود.

زمان‌بندی دقیق و حلقه ارتباطی Real-Time:

تمامی دستگاه‌ها به صورت هماهنگ در بازه‌های زمانی از پیش تعریف شده ارتباط برقرار می‌کنند، که باعث جلوگیری از ارسال داده‌های غیرهمزمان می‌شود.

ساختار تعیین موقعیت داده‌ها در فریم:

هر دستگاه فقط بخش مربوط به خود را در فریم تغییر می‌دهد و نیازی به ارسال پاسخ جداگانه وجود ندارد، بنابراین هیچ برخوردی رخ نمی‌دهد.

آدرس دهی در این پروتکل

در شبکه EtherCAT، آدرس دهی دستگاه های Slave به روش های زیر انجام می شود:

آدرس دهی خودکار (Auto-Increment Addressing):

در این روش، Master هنگام راه اندازی شبکه به صورت خودکار به هر دستگاه یک آدرس موقعیتی اختصاص می دهد. هر بسته داده ای که در گذرگاه ارسال می شود، از اولین دستگاه عبور کرده و هر دستگاه فقط بخش مربوط به خود را تغییر می دهد. این روش نیازی به پیکربندی دستی آدرس ها ندارد.

آدرس دهی ثابت (Configured Addressing):

هر دستگاه یک آدرس ثابت دارد که می توان آن را به صورت دستی تنظیم کرد. این آدرس ها به Master کمک می کنند که به صورت مستقیم با هر دستگاه خاص ارتباط برقرار کند.

آدرس دهی مبتنی بر اترنت استاندارد (Ethernet MAC Addressing):

دستگاه های EtherCAT دارای آدرس MAC اترنت استاندارد هستند، اما این آدرس در عملکرد شبکه EtherCAT معمولاً مورد استفاده قرار نمی گیرد.

مسیریابی در این پروتکل

پروتکل EtherCAT نیازی به مسیریابی پیچیده ندارد، زیرا تمام دستگاه های Slave در یک توپولوژی خطی (Daisy Chain) یا حلقه ای (Ring Topology) متصل شده اند. مسیریابی به شکل زیر مدیریت می شود:

ساختار زنجیره ای و پردازش مستقیم فریم ها:

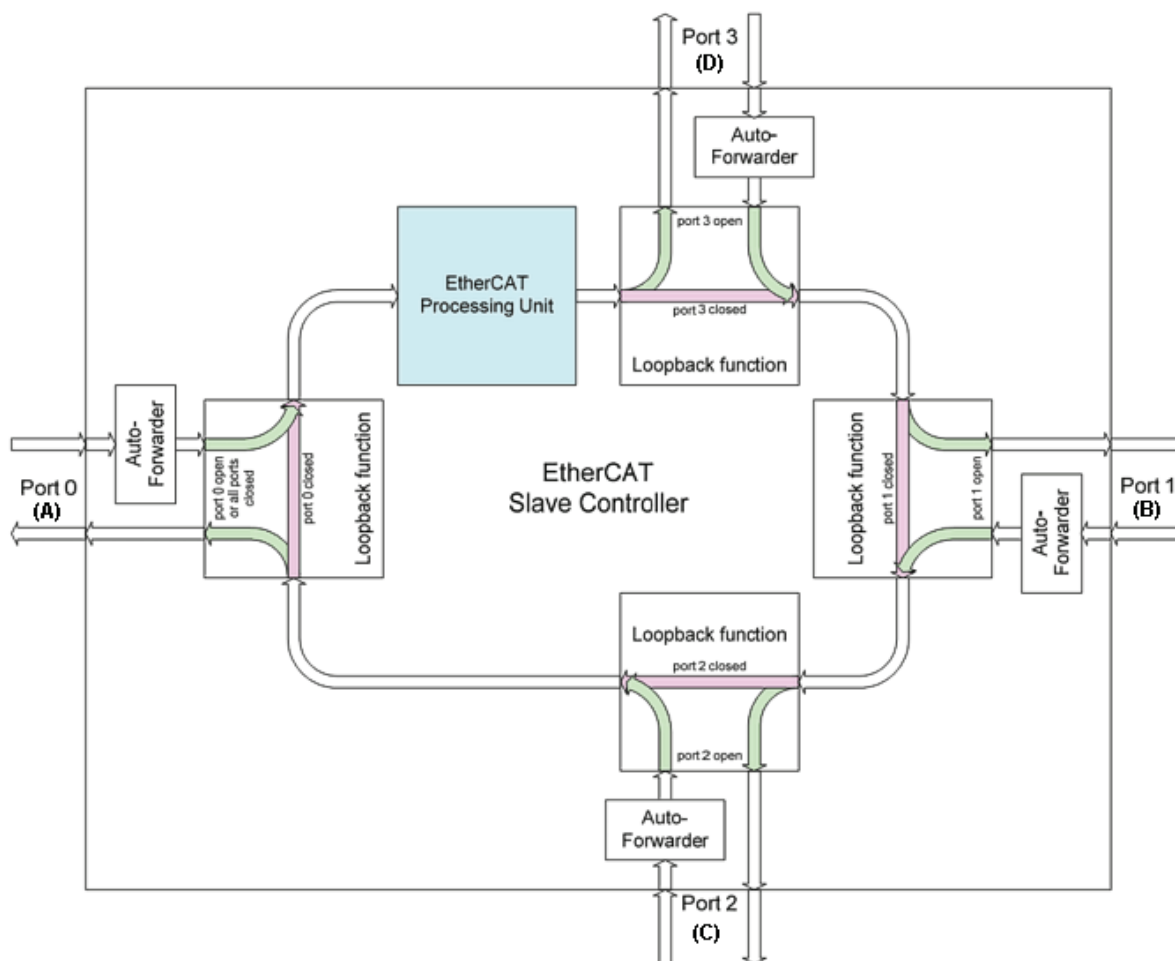
هر بسته داده ای از تمامی دستگاه های موجود در مسیر عبور می کند. هر دستگاه Slave فقط داده های مربوط به خود را پردازش کرده و فریم را به دستگاه بعدی ارسال می کند.

مکانیزم Loopback در صورت نیاز:

در صورت خرابی یک مسیر، شبکه می تواند از مسیرهای جایگزین استفاده کند. همان طور که در تصویر مشاهده می شود، تابع Loopback در EtherCAT Slave Controller امکان هدایت داده ها به مسیرهای مختلف را فراهم می کند.

چرا EtherCAT نیازی به مسیریابی پیچیده ندارد؟

ساختار Master-Slave: تنها یک Master در شبکه وجود دارد که کنترل ارتباطات را بر عهده دارد. عبور فریم از همه دستگاه ها: به جای ارسال بسته های داده جداگانه، یک فریم واحد از تمامی دستگاه ها عبور کرده و هر دستگاه اطلاعات مربوط به خود را پردازش می کند. در شبکه های سنتی مانند TCP/IP، برای یافتن مسیرهای بهینه نیاز به جداول مسیریابی است، اما در EtherCAT، مسیر داده از پیش تعیین شده و مستقیم است.



شکل ۳ یک EtherCAT Slave Controller نمایش داده شده است که دارای چهار پورت (Port 0 تا Port 3) است.

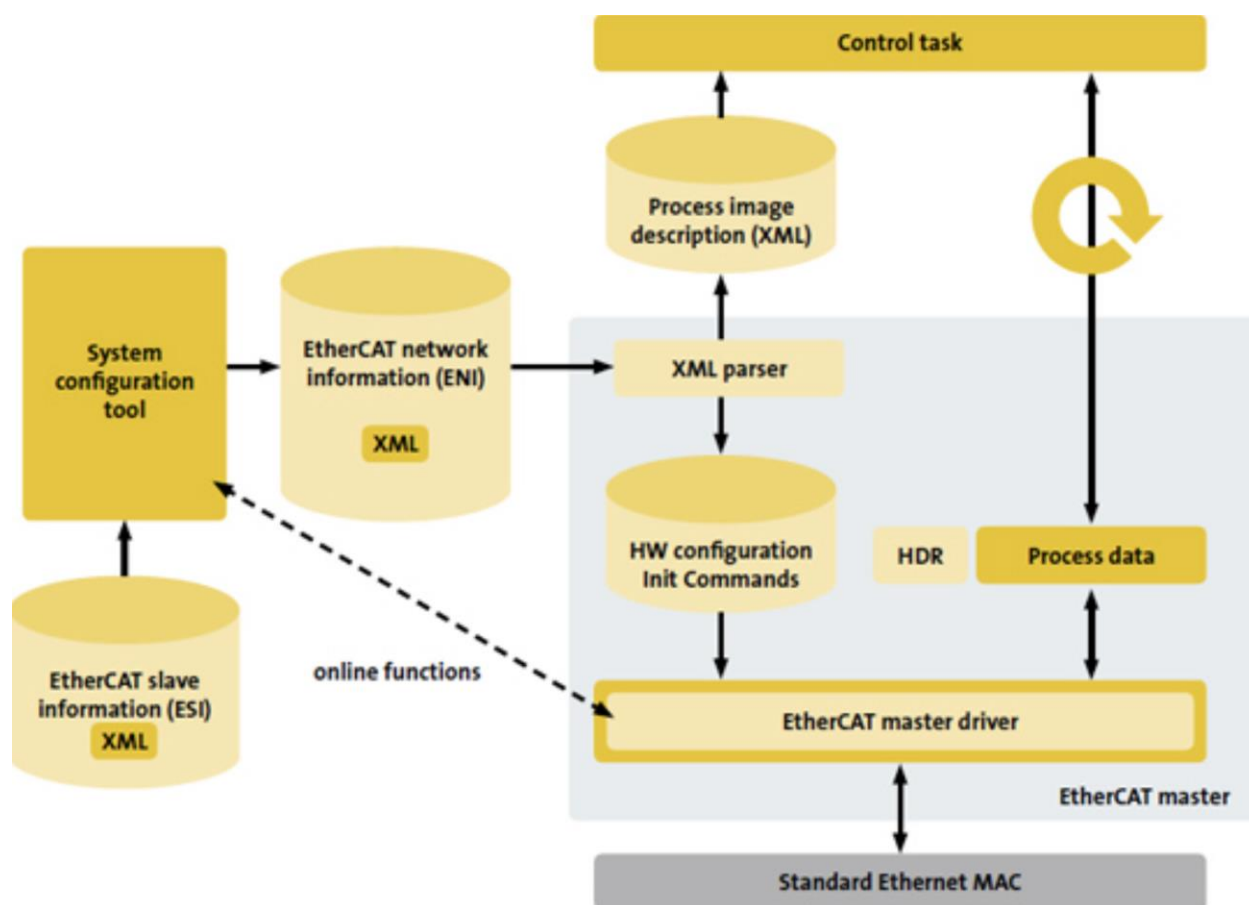
توضیحات مربوط به شکل ۳

در این تصویر برای درک بهتر از آدرس دهی و مسیریابی ارائه شده است که جزئیات آن به صورت زیر است:

- ❖ Port ۰ (A): ورود داده از Master
- ❖ Port ۱ (B): ارسال داده به دستگاه‌های دیگر
- ❖ Port ۲ (C): مسیر جایگزین برای ارسال داده
- ❖ Port ۳ (D): مسیر کمکی دیگر برای ارسال داده

هر دستگاه می‌تواند داده‌ها را از یک پورت دریافت کرده و از طریق دیگر پورت‌ها ارسال کند، در صورتی که مسیر پیش‌فرض مسدود یا غیرفعال شود در این ساختار، آدرس‌دهی به‌صورت ترتیبی انجام شده و نیازی به تنظیم دستی مسیرها وجود ندارد.

نحوه مدیریت جریان داده در این پروتکل



شکل ۴) مدیریت جریان داده در پروتکل EtherCAT به صورت دیاگرام نمایش داده شده است.

مراحل مدیریت جریان داده به صورت زیر است:

۱. تنظیمات اولیه و پیکربندی شبکه

System Configuration Tool (ابزار پیکربندی سیستم) مسئول تولید اطلاعات مربوط به شبکه EtherCAT است. این ابزار از فایل‌های EtherCAT Slave Information (ESI) که به صورت XML هستند، برای شناسایی و تنظیم تجهیزات متصل به شبکه استفاده می‌کند. پس از انجام پیکربندی، EtherCAT Network Information (ENI) در قالب XML ایجاد می‌شود که شامل اطلاعات توپولوژی شبکه و پارامترهای ارتباطی است. همچنین، این ابزار از Online Functions برای برقراری ارتباط و نظارت بر وضعیت سیستم استفاده می‌کند.

۲. پردازش اطلاعات و ارسال به Master

ENI (EtherCAT Network Information) به یک XML Parser ارسال می‌شود. این واحد مسئول تجزیه اطلاعات و آماده‌سازی آنها برای اجرا است. HW Configuration Init Commands (دستورات پیکربندی سخت‌افزار) از این اطلاعات استخراج شده و به EtherCAT Master Driver ارسال می‌شوند.

۳. اجرای دستورات و پردازش داده‌ها

EtherCAT Master Driver اطلاعات دریافت‌شده را پردازش کرده و از طریق Standard Ethernet MAC برای ارسال روی شبکه آماده می‌کند. در این مرحله، داده‌های فرآیند از طریق HDR (Header Processing) مدیریت شده و به‌عنوان Process Data آماده استفاده در برنامه‌های کنترلی می‌شود.

۴. اجرای وظایف کنترلی و تبادل داده

داده‌های پردازش‌شده توسط Process Data به Control Task (وظیفه کنترلی) ارسال می‌شود. Control Task مسئول اجرای الگوریتم‌های کنترلی، تصمیم‌گیری و ارسال پاسخ‌های مناسب به سیستم است. در صورت نیاز، اطلاعات پردازشی در Process Image Description (XML) ذخیره شده و مجدداً به سیستم بازگردانده می‌شود.

۵. ارسال داده به شبکه و ارتباط با Slaveها

EtherCAT Master Driver داده‌های پردازش‌شده را از طریق Standard Ethernet MAC روی شبکه EtherCAT ارسال می‌کند. این داده‌ها به‌صورت Frameهای EtherCAT در طول خط ارسال شده و توسط تجهیزات Slave دریافت و پردازش می‌شوند. هر Slave اطلاعات موردنظر خود را از فریم‌ها استخراج کرده و پاسخ مناسب را در همان فریم قرار می‌دهد تا به Master بازگردد.

مدیریت و تشخیص خطا در لایه‌های مختلف این پروتکل

پروتکل EtherCAT تشخیص خطا را در چندین لایه مدل OSI پیاده‌سازی کرده است. تشخیص خطا را در لایه‌های مختلف بررسی می‌کنیم:

لایه فیزیکی (Physical Layer)

در این لایه، خطاهای ناشی از نویز، قطعی یا کاهش کیفیت سیگنال در کابل‌های اترنت رخ می‌دهند. مکانیزم‌های تشخیص خطا در این لایه شامل موارد زیر است:

- ❑ کدینگ خطی (Line Coding) در استاندارد (۱۰۰ BASE-TX) Fast Ethernet برای بررسی سلامت سیگنال
- ❑ بررسی پیوستگی لینک (Link Integrity Check) برای تشخیص قطع شدن کابل
- ❑ مکانیزم Auto-Negotiation برای اطمینان از سازگاری سرعت ارتباط (در برخی تنظیمات)
- ❑ BER (Bit Error Rate) Monitoring برای بررسی نرخ خطای بیت

مثال

در صورتی که کابل دچار قطعی شود، دستگاه Master این خطا را تشخیص داده و آدرس تجهیزات Slave را مجدداً تنظیم می‌کند.

لایه پیوند داده (Data Link Layer)

این لایه مهم‌ترین لایه برای تشخیص خطا در EtherCAT است و شامل مکانیزم‌های حفاظتی قوی برای تشخیص و اصلاح خطاهای داده‌ای می‌شود. روش‌های تشخیص خطا:

❑ CRC-۱۶ (Cyclic Redundancy Check) برای تشخیص خطای انتقال فریم‌های EtherCAT

❑ Lost Frame Detection (تشخیص فریم‌های گم‌شده)

❑ Watchdog Mechanism برای تشخیص تاخیر بیش از حد در پاسخ دستگاه‌های Slave

❑ Automatic Retransmission Mechanism برای ارسال مجدد داده‌ها در صورت بروز خطا

مثال

در صورت خراب شدن یک فریم داده، فریم دوباره ارسال می‌شود یا دستگاه Master یک پیام خطا دریافت می‌کند.

لایه شبکه (Network Layer)

از آنجایی که EtherCAT یک پروتکل بدون مسیریابی (Non-Routable Protocol) است و فقط در یک شبکه محلی استفاده می‌شود، مکانیزم‌های تشخیص خطا در این لایه محدود هستند. با این وجود، برخی مکانیزم‌های بررسی وضعیت شبکه به کار می‌روند:

❑ Topology Checking (بررسی توپولوژی) برای تشخیص تغییرات ناخواسته در شبکه

❑ Error Counters in Slaves (شمارنده‌های خطا در تجهیزات Slave) برای مانیتورینگ بسته‌های از دست‌رفته

مثال

اگر یکی از تجهیزات Slave دچار مشکل شود، Master این خطا را در ENI File (EtherCAT Network Information) ثبت می‌کند.

لایه انتقال (Transport Layer)

در EtherCAT هیچ پروتکل انتقالی مانند TCP یا UDP وجود ندارد، اما روش‌هایی برای تشخیص مشکلات ارتباطی دارد:

❑ Working Counter Mechanism که تعداد فریم‌های پردازش‌شده توسط هر Slave را بررسی می‌کند. اگر مقدار این شمارنده

با مقدار موردانتظار Master متفاوت باشد، نشان‌دهنده یک خطای ارتباطی است.

❑ Redundant Ring Topology Support که اجازه می‌دهد در صورت قطع شدن مسیر اصلی، داده‌ها از مسیر جایگزین عبور

کنند.

مثال

اگر یکی از فریم‌های EtherCAT به درستی توسط یک دستگاه Slave پردازش نشود، مقدار Working Counter تغییر نخواهد کرد و Master متوجه خطا می‌شود.

لایه کاربردی (Application Layer)

در این لایه، خطاها معمولاً به دلیل پیکربندی نادرست، خطای نرم‌افزاری یا مشکلات در برنامه کنترلی رخ می‌دهند. مکانیزم‌های تشخیص خطا:

❑ State Monitoring (نظارت بر وضعیت تجهیزات) برای تشخیص خطای عملکردی هر Slave

- ❌ Sync Manager Error Handling برای جلوگیری از ناهماهنگی در انتقال داده‌ها
- ❌ CoE (CANopen over EtherCAT) Emergency Messages که در صورت وقوع یک رویداد غیرمنتظره ارسال می‌شود.

مثال

اگر یک دستگاه Slave مقدار داده‌های ارسال شده را در زمان مناسب نداشته باشد، خطای Sync Manager Error رخ می‌دهد.

انواع پیام در این پروتکل

پیام‌های فرآیند داده‌ها (Process Data Messages - PDM)

این پیام‌ها برای ارسال و دریافت داده‌های I/O از دستگاه‌های Slave استفاده می‌شوند. دارای تاخیر کم و مناسب برای کنترل Real-Time. داده‌ها در مدل PDO (Process Data Object) سازمان‌دهی می‌شوند. درون Datagram‌ها قرار گرفته و مستقیماً پردازش می‌شوند.

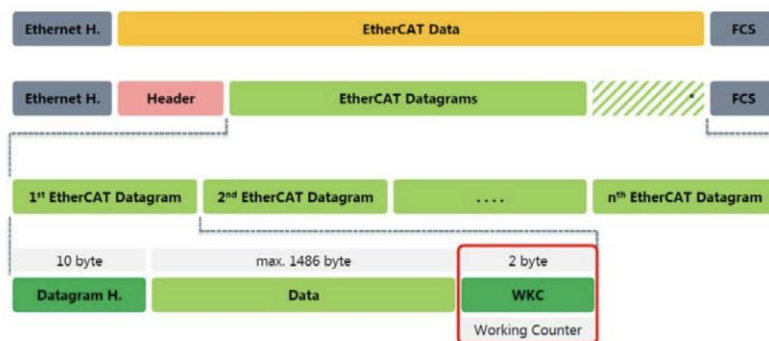
پیام‌های پارامتری و تنظیمات (Service Data Messages - SDM)

این پیام‌ها برای پیکربندی، خواندن یا نوشتن پارامترها در دستگاه‌های Slave استفاده می‌شوند. از مدل SDO (Service Data Object) تبعیت می‌کنند. بیشتر برای تنظیمات اولیه و تغییر پارامترها استفاده می‌شوند و زمان پاسخگویی کمتر بحرانی است.

پیام‌های دسترسی به حافظه (Mailbox Messages)

برای ارتباطات غیر همزمان و ارسال داده‌های اضافی مانند پروتکل‌های CAN over EtherCAT و Ethernet over EtherCAT (EoE) استفاده می‌شوند. شامل پیام‌های هشدار، ارسال لاگ‌ها، تنظیمات پیچیده‌تر. به دلیل زمان پاسخگویی بالاتر، برای کنترل بلادرنگ استفاده نمی‌شوند.

فرمت کلی پیام‌ها در این پروتکل



شکل ۵) در این تصویر می‌توان بخش‌های اصلی یک پیام EtherCAT را مشاهده کرد.

توضیحات مربوط به شکل ۵:

Ethernet Header (Ethernet H.)

این قسمت، هدر استاندارد Ethernet را نشان می‌دهد که برای شناسایی بسته‌های اترنت استفاده می‌شود.

EtherCAT Data

این بخش شامل داده‌های EtherCAT است که در قالب چندین Datagram سازمان‌دهی شده‌اند.

Header

این قسمت شامل اطلاعات کنترلی مربوط به هر EtherCAT Datagram است که تعیین می‌کند این بسته به کدام دستگاه‌ها مربوط می‌شود.

EtherCAT Datagrams

هر فریم EtherCAT شامل چندین Datagram است که به صورت متوالی پردازش می‌شوند.

Datagram Header (Datagram H.)

هر Datagram دارای یک هدر ۱۰ بایتی است که شامل اطلاعاتی مانند نوع دسترسی و آدرس دستگاه است.

Data

این قسمت شامل داده‌هایی است که در هر Datagram ارسال یا دریافت می‌شوند.

WKC (Working Counter)

این فیلد ۲ بایتی در هر Datagram برای بررسی صحت ارسال و دریافت داده‌ها استفاده می‌شود. مقدار WKC هنگام عبور از هر نود در شبکه افزایش می‌یابد و می‌توان از آن برای تشخیص خطاهای ارتباطی استفاده کرد.

FCS (Frame Check Sequence)

این فیلد برای بررسی یکپارچگی داده‌ها و تشخیص خطاهای انتقال مورد استفاده قرار می‌گیرد.

رویکرد تشخیص خطا

EtherCAT به گونه‌ای طراحی شده است که خطاها را به حداقل برساند. چندین رویکرد در تشخیص خطا داریم:

ساختار مبتنی بر Master/Slave:

در این پروتکل، فقط Master فریم‌ها را ارسال می‌کند و Slave‌ها فقط داده‌ها را ویرایش کرده و ارسال می‌کنند، که احتمال بروز خطا را کاهش می‌دهد.

استفاده از مکانیسم CRC در لایه Data Link:

برای بررسی صحت داده‌های ارسالی، از Frame Check Sequence (FCS) و Cyclic Redundancy Check (CRC) استفاده می‌شود.

شماره‌گذاری بسته‌ها با Working Counter (WKC):

WKC به ازای هر Slave افزایش می‌یابد و در صورت عدم تطابق مقدار انتهایی، خطا تشخیص داده می‌شود.

رویکرد تصحیح خطا

EtherCAT برخلاف پروتکل‌هایی مانند TCP/IP که دارای مکانیسم بازارسال (Retransmission) در لایه Transport هستند، بر تشخیص سریع خطا و ارسال مجدد به صورت خودکار توسط Master متکی است.

بازارسال خودکار پیام توسط Master

در صورت بروز خطا، Master می‌تواند به سرعت همان بسته را مجدداً ارسال کند. از آنجایی که EtherCAT نیازی به تاییدیه (ACK) از Slaveها ندارد، این فرآیند سریع‌تر از مکانیزم بازارسال در پروتکل‌های دیگر مانند TCP انجام می‌شود.

استفاده از Redundancy برای جلوگیری از خطا

❖ Redundant Master: در صورت خرابی Master، یک Master جایگزین وارد عمل می‌شود.

❖ Redundant Communication Path: مسیر جایگزین برای ارتباطات در صورت قطع شدن لینک اصلی استفاده می‌شود.

بررسی و تصحیح خطای بیت (Bit Error Handling)

برخی از کنترلرهای EtherCAT قابلیت شناسایی و تصحیح خطای بیت (Single Bit Error) را دارند که باعث جلوگیری از خرابی کلی فریم داده می‌شود.

منابع

https://www.ethercat.org/en/why_use_ethercat.htm

<https://www.ethercat.org/en/technology.html>

<https://dewesoft.com/blog/what-is-ethercat-protocol>

https://www.ethercat.org/2011/italy/download/02_Ethernet_Intro_ETG.pdf

https://en.wikipedia.org/wiki/Ethernet_physical_layer

https://www.ethercat.org/download/documents/ethercat_diagnosis_for_users.pdf