EtherCAT® and EtherCAT® P
Slave Implementation Guide

Document: ETG.2200 V3.1.0

Nomenclature:

ETG-Number	ETG.2200

SECTION I – EtherCAT Slave Introduction and Implementation Procedure

SECTION II – EtherCAT Development Components

SECTION III – EtherCAT P Introduction and Implementation

Created by:	EtherCAT Technology Group
Contact:	info@ethercat.org
Date:	23.11.2018

EtherCAT® Technology Group

LEGAL NOTICE

Trademarks and Patents

EtherCAT®, Safety over EtherCAT® and EtherCAT P® are registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Disclaimer

The documentation has been prepared with care. The technology described is, however, constantly under development. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Copyright

DOCUMENT HISTORY

Version	Comment
1.0.0	Official Release
1.1.0	Document revised Editorial Changes ESC Variants updated More Implementation Products added New Documentation Links
1.1.1	Editorial Changes
1.1.2	Use of Marking Rules and Indicator Specification added
1.1.3	Editorial Changes
1.1.4	Editorial Changes
1.1.5	Editorial Changes Conformance Test Policy added Vendor ID Policy added Minor changes in Step by Step Implementation
1.1.6	Editorial Changes
2.0.0	Document revised New document structure Enhanced General Procedure - Step by Step Major content enhancements
2.0.1	Editorial Changes Policies added Download links updated/fixed Minor content enhancements
2.0.3	Editorial Changes
2.1.0	Contact email address is conformance@ethercat.org
2.1.1	Development products updated
2.1.2	Development products (Renesas, TESSERA) updated
2.1.4	Correction of trade mark term acc. To ETG.9001
2.1.5	Editorial Changes
2.1.6	Added reference to ETG.5003-2
2.1.7	Editorial Changes, updates of hyperlinks and ESC overview Add and update chapters on Knowledge Base/Search/Download, support
3.0.0	Add EtherCAT P section
3.0.1	Development products (profichip) updated
3.0.2	Add description on CTT license
3.0.3	Update of EtherCAT P specifications status (available as release)
3.0.4	Update of contact graphics Update of colors of some graphics
3.1.0	Add “Evaluate your current EtherCAT Slave for EtherCAT P” Add products Update abbreviations, links, references, layout

DOCUMENT ORGANIZATION

This document describes how to accomplish a successful EtherCAT® slave implementation from a generic and a practical point of view. It answers the following questions:

How is the EtherCAT slave architecture?

What steps have to be done to implement an EtherCAT slave?

Which documents are available for a successful device implementation?

What kinds of EtherCAT development components are available? What are the differences?

Is EtherCAT training available?

Is technical support available?

Why to attend a Plug Fest?

How to obtain conformance for EtherCAT devices?

There are many possibilities how an EtherCAT slave implementation can be done. However, the way it is described in this document has proofed many times to lead fast to an EtherCAT slave device implementation. The document is organized in two sections:

SECTION I – EtherCAT Slave Introduction and Implementation Procedure

Section I deals with principal aspects of an EtherCAT slave implementation. Chapter 0 provides a brief EtherCAT technology background focusing the slave. In chapter 2 the implementation steps for a slave device are described, containing slave implementation criteria and a list of useful tools. This document gives some implementation notes, too.
Chapter 3 describes support and trainings, which is provided by the EtherCAT Technology Group.

SECTION II – EtherCAT Development Components

Section II contains device specific descriptions for further implementation aspects. An overview to available evaluation boards is given in chapter 2. Following is a list of available EtherCAT Communication Modules in chapter 3, and Slave Controllers (ESCs) in chapter **Error! Reference source not found.**.

SECTION III – EtherCAT P Introduction and Implementation

Section III gives an overview of the EtherCAT enhancement “EtherCAT P”, including all related implementation topics.

® EtherCAT is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

EtherCAT® and EtherCAT® P Slave Implementation Guide

SECTION I – EtherCAT Slave Introduction and Implementation Procedure

Technology overview, Network Architecture and Functionality, Slave Implementation Procedure, Exemplary Implementation, Support and Training, EtherCAT Technology Group

EtherCAT® Technology Group

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

Abbreviations	Description
µC / MC	Microcontroller, host controller, application controller
ADS	Automation Device Specification
AL	Aplication Layer
AoE	ADS over EtherCAT
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
CiA	CAN in Automation
CoE	CAN application protocol over EtherCAT
CPU	Central Processing Unit
DC	Distributed Clocks
DL(L)	Data Link Layer
DPRAM	Dual Ported Random Access Memory
DuT	Device under Test
EEPROM	Electrical Erasable Programmable Read Only Rom
ENI	EtherCAT Network Information (Network configuration in XML format)
EoE	Ethernet over EtherCAT
EPU	EtherCAT Processing Unit
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information (device description in XML format)
ESM	EtherCAT State Machine
ETC	EtherCAT Test Center
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automation Technology
FCS	Frame Check Sequence
FMMU	Fieldbus Memory Management Unit
FoE	File Access over EtherCAT
FPGA	Field Programmable Gate Array
FSoE	Fieldbus Safety over EtherCAT
GPIO	General Purpose I/O
HAL	Hardware Abstraction Layer
I^{2}C	Inter-Integrated Circuit
LED	Light Emitting Diode
LVDS	Low Voltage Differential Signaling
MCI	Micro Controller Interface
MDP	Modular Device Profile
MII	Media Independent Interface
NIC	Network Interface Controller
NVRAM	Non Volatile Random Access Memory
NW	NetWork
OEM	Original Equipment Manufacturer
PD	Power Device
PDI	Process Data Interface
PDO	Process Data Object
PHY / PL	PHYsical Layer
PIC	Programmable Integrated Circuit
PLC	Programmable Logic Controller
PSD	Power Sourcing Device
RMII	Reduced Media Independent Interface
SDO	Service Date Object
SII	Slave Information Interface
SM	SyncManager
SoE	Servo drive over EtherCAT

Abbreviations	Description
SPI	Serial Peripheral Interface
SSC	Slave Stack Code
TC	Technical Committee
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
XML	Extended Mark-up Language

1. Introduction

This chapter presents a brief overview to the EtherCAT architecture and technology. Since EtherCAT technology covers more details than presented here, a list of documents which provide deeper understanding of the technology is given first. Corresponding text passages in this guide refer to these documents. In the following subsections the basic system architecture and the system functionality of an EtherCAT network is described. Since this is a slave implementation guide, it focuses on the slave.

1.1. Documents for Detailed Information and Further Reading

It is recommended to consider the following information before proceeding to develop an EtherCAT device. Some of the information below is provided in the member area[1] of the website of the EtherCAT Technology Group (ETG). ETG membership is free of charge and is required to access the wide range of EtherCAT related documents, specifications and guidelines, as well as to receive technical support from the ETG. See chapter 2.2.2 for how to become a member and to get an account.

The complete list of all available EtherCAT documentation can be found at the download section of the ETG website (www.ethercat.org/en/downloads.html). Table 1 lists documents related to slave implementation and general EtherCAT technology overview.

Table 1: EtherCAT Information, Standards and References

	Subject	Documents, Description and Access
Introduction	Brochures and Presentations	EtherCAT is introduced in several brochures, published in different languages: → English | Japanese | Chinese | German | Korean | Italian | Spanish This description of EtherCAT technology basics is an introduction in → English | Japanese | Chinese | German | French | Italian | Portuguese An Introduction to Safety over EtherCAT is available in → English | German
	Articles	EtherCAT has been introduced in several articles. A selection of them is given here. → PC Control (English): 04/2009 | 11/2003 | 06/2003 → Elektronik 23/03 (German) → AUTlook 2-3/05 (German)
	Videos	Functional principle of the frame processing order and data exchange: → https://www.youtube.com/user/EtherCATGroup
Detailed Reading	Knowledge Base	An online information system contains FAQs and EtherCAT feature descriptions. → www.ethercat.org/kb[1]
	Technology Description	Section I of the Beckhoff EtherCAT Slave Controller Datasheet ET1100 contains a comprehensive description of EtherCAT functionality. Sections II (ESC register description) and section III (hardware specification) provide more detailed Information. → beckhoff.org > Download > Documentation > EtherCAT Development Products
	Proceedings of ETG Events	Minutes of the Technical Committee Meetings hold actual technology development topics: → www.ethercat.org → Downloads → Select Filter: Proceedings and Papers → Technical Committee Meeting
	Slave	The reference on the ETG website to this document:
Development	Communication Slides	The communication slides provide a broad description of EtherCAT mechanisms for developers. → English[1] | Japanese[1]
	PHY Selection Guide	The PHY Selection Guide contains information for physical level connection components of several vendors that are available for EtherCAT communication. → EtherCAT PHY Selection Guide

[1] ETG membership sign-in required.

Specifications	Communication Specification	EtherCAT is specified by the EtherCAT Communication specification ETG.1000 parts 2 to 6. → ETG.1000 series: www.ethercat.org/etg1000[1] Note ETG.1000 represents the IEC 61158 - Type 12 (EtherCAT).
	EtherCAT Slave Information (ESI)	The EtherCAT Slave Information File (ESI) is the EtherCAT device description in XML format. It is defined in the ETG.2000 ESI specification. Device description example files can also be found here. The ETG.2001 ESI Annotation also contains sample files for ESI file development. → ETG.2000: www.ethercat.org/etg2000[1]
	Safety over EtherCAT	Safety over EtherCAT defines a protocol layer for safe data exchange. ETG.5100 contains the safety protocol and ETG.6100 specifies a Safety Drive Profile. → ETG.5100: www.ethercat.org/etg5100[1] → ETG.6100: www.ethercat.org/etg6100[1] Note ETG.5100 represents the IEC 61784 international standard.
	Drives	The implementation directive for the CiA402 Drive Profile is defined by the ETG.6010 specification. → ETG.6010: www.ethercat.org/etg6010[1] Note ETG.6010 is based on the IEC 61800-7-201 (CiA402 drive profile).
	Conformance	Conformance Test rules are defined in the EtherCAT Conformance Test Policy. The Conformance Guide describes how developers can obtain conformance (ETG.7000). Additionally, a Test Record and the Test Request form are available here. → ETG.7000x: www.ethercat.org/etg7000[1]
	Firmware Update	ETG.5003.2 Firmware Update Specification This specification is mandatory only for devices supporting the profile number 5003 (Semi Device Profile), however, it is a good guideline for a firmware update implementation on any EtherCAT slave → ETG.5003-2: www.ethercat.org/etg5003
	Trademark, Logo and Labelling Rules	Marking rules, trademark, logo and labelling usage for products and documentations applying EtherCAT technology or referring to it are defined in the ETG.1300 and the ETG.9001 specifications: → ETG.1300: www.ethercat.org/etg1300[1] → ETG.9001: www.ethercat.org/etg9001[1]

1.2. EtherCAT System Architecture

The basic EtherCAT system configuration is shown in Figure 1. The EtherCAT master uses a standard Ethernet port and network configuration information stored in the EtherCAT Network Information file (ENI). The ENI is created based on EtherCAT Slave Information files (ESI) which are provided by the vendors for each device. Slaves are connected via Ethernet, any topology type is possible for EtherCAT networks.

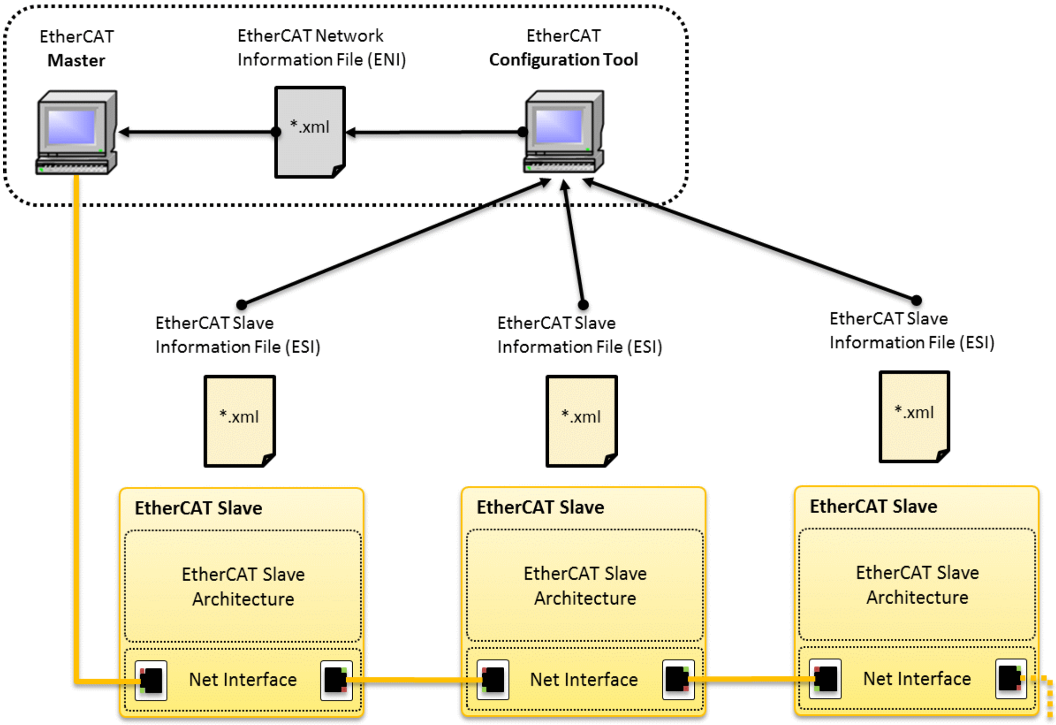


Figure 1: EtherCAT Network Architecture

1.2.1. Configuration Tool

- EtherCAT Configuration Tool
  The Configuration Tool is used to generate network description, the so called EtherCAT Network Information file (ENI, XML file based on a pre-defined file schema). This information is based on the information provided by the EtherCAT Slave Information files (ESI, device description in XML format, see chapter 1.2.3) and/or the online information provided by the slaves in their EEPROM and their object dictionaries.

- EtherCAT Network Information File (ENI)
  The ENI file describes the network topology, the initialization commands for each device and the commands which have to be sent cyclically. The ENI file is provided to the master, which sends commands according to this file. For more information see ETG.2100 EtherCAT Network Information (ENI) Specification.

1.2.2. Master System

- Hardware: The only hardware requirement for an EtherCAT master is a standard Network Interface Controller (NIC, 100 MBit/s Full duplex).

- Software: A real time runtime environment drives the slaves in the network. Since this guide focuses on the slave, it won't get into detail to master software. Further information is available at the ETG website's product section.

1.2.3. Slave Device

Figure 2 shows the EtherCAT network with focus on the slave architecture. Basically, the slave contains three main components:

- Physical Layer (PL):	Network interface
- Data Link Layer (DL):	EtherCAT Slave Controller (ESC, communication module) and EEPROM
- Application Layer (AL):	Host Controller (also called application controller or microcontroller, µC)

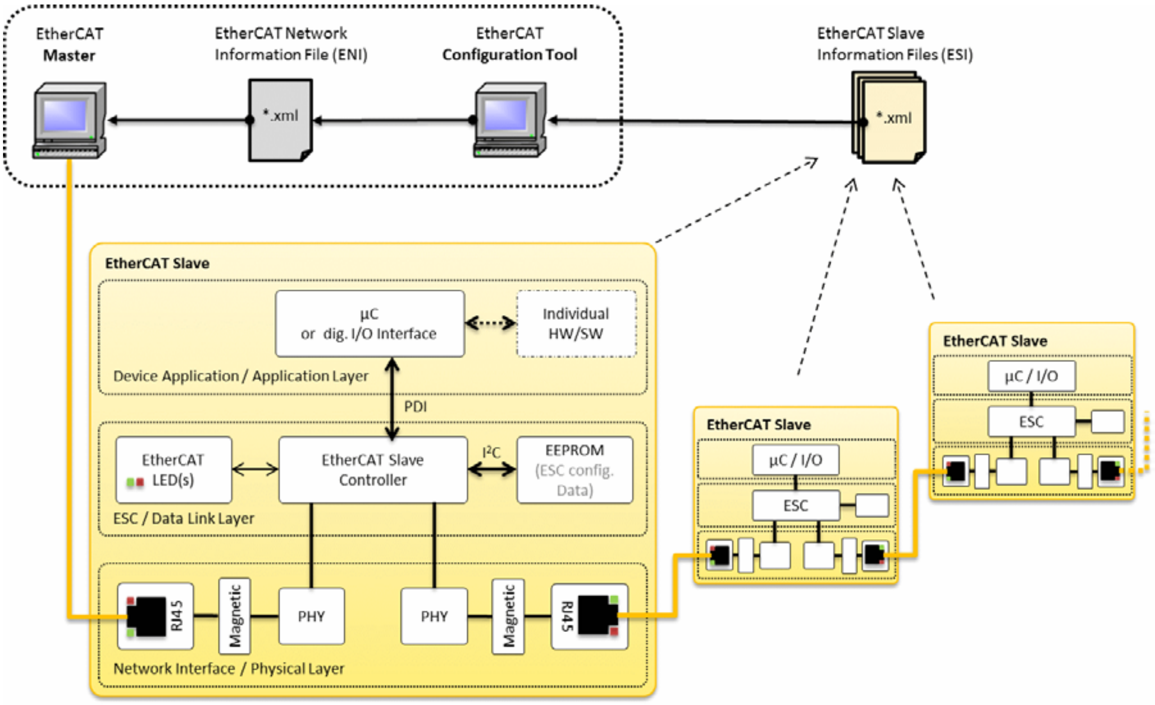

EtherCAT Master
EtherCAT Network Information File (ENI)
EtherCAT Configuration Tool
EtherCAT Slave Information Files (ESI)
*.xml
*.xml
EtherCAT Slave
µC or dig. I/O Interface
Individual HW/SW
Device Application / Application Layer
PDI
EtherCAT LED(s)
EtherCAT Slave Controller
I²C
EEPROM (ESC config. Data)
ESC / Data Link Layer
RJ45
Magnetic
PHY
PHY
Magnetic
RJ45
Network Interface / Physical Layer
EtherCAT Slave
µC / I/O
ESC
EtherCAT Slave
µC / I/O
ESC

Figure 2: EtherCAT Slave Architecture

In detail, the slave consists of the following components. Criteria for these components concerning the device design and development are discussed in chapter 2.3.

- **Standard Ethernet Physical Layer Components (Network Interface)**

The network interface contains the physical layer (PHY) components to process fieldbus signals. It forwards network data to the slave controller (ESC) and applies signals from the ESC to the network. The physical layer is based on the standards defined by standard Ethernet (IEEE 802.3).

i. Plugs:
Ethernet cable connectors. Standard RJ45 connectors (recommended) or M12 D-code connectors can be used. As EtherCAT cables, shielded twisted pair enhanced category 5 (CAT 5e STP) or better is recommended. Select an appropriate cable for the environment where the machine is installed.

ii. Magnetics:
Pulse transformers for galvanic isolation.

iii. Standard PHYs:
A chip that implements the hardware functions for sending and receiving Ethernet frames. It interfaces to the line modulation at one end and binary packet signaling at the other. Refer to the PHY Selection Guide for details.

E-Bus (LVDS) does not provide galvanic isolation, electromechanical connections are vendor specific. For external cable-based connections 100BASE-TX is recommended.

- **EtherCAT Slave Controller (ESC) and Process Data Interface (PDI)**

The *ESC* is a chip for EtherCAT communication. The ESC handles the EtherCAT protocol in real-time by processing the EtherCAT frames on the fly and providing the interface for data exchange between EtherCAT master and the slave's local application controller via registers and a Dual-Port-RAM (DPRAM).

The ESC can either be implemented as FPGA (Field Programmable Gate Array) or as ASIC (Application Specific Integrated Circuit). The EtherCAT frame is completely processed by the ESC, and hence, the processing speed is basically the same for any EtherCAT slave. It does not depend on the host controller performance. At the same time, the application on the host controller does not need to process the Ethernet frame forwarding, and hence, the host controller resources are free to be used by the slave's application. The processing frequency of the host controller is defined by the host application.

The ESC processes EtherCAT frames on the fly and provides data for a local host controller or digital I/Os via the Process Data Interface (PDI). Different ESCs might support different PDIs, most common PDIs are: ESC.:

i. Up to 32 Bit digital I/O

ii. Serial Peripheral Interface (SPI)

iii. 8/16-bit synchronous/asynchronous Microcontroller Interface (MCI)

iv. With FPGA: specific on-board-bus (Avalon on Altera devices resp. OPB on Xilinx devices)

Process data and parameters are exchanged via a DPRAM in the ESC. To ensure data consistency appropriate mechanisms are provided by the ESC hardware (defined by the EtherCAT protocol, e.g. SyncManager (SM), chapter 1.3.4).

In case of an FPGA implementation, the ESC is realized as IP core to enable EtherCAT communication and application-specific functions. The EtherCAT device functionality is configurable with regard to the EtherCAT features such as number of Fieldbus Memory Management Units (FMMUs) and SyncManagers, DC support, PDI (chapter 2.3).

FPGA implementations are at option in two ways. One way is integrating ESC and a soft core µC on the FPGA. As PDI the FPGA on-board bus is then used. Another option is using the FPGA solely for the ESC functionality and connecting an external µC via µC/SPI, s. Figure 3.

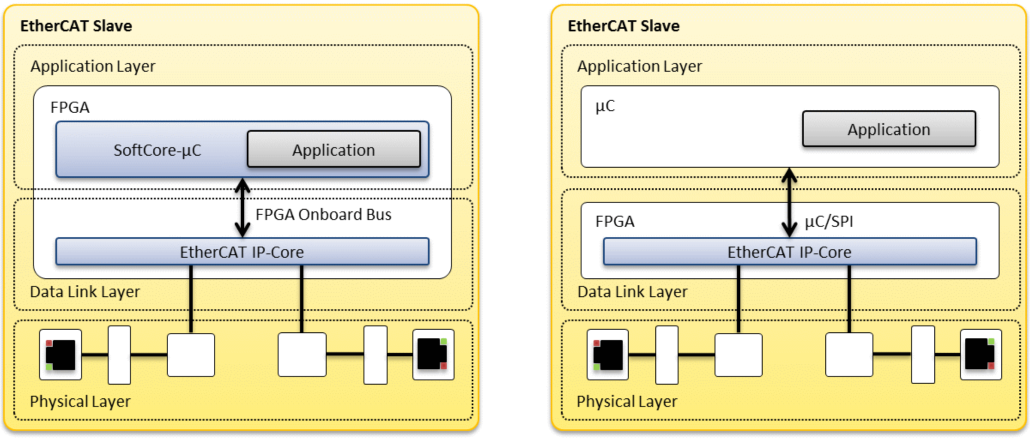


Figure 3: FPGA Implementations of an EtherCAT Slave

A plug-in for Altera or Xilinx development environments is available to configure the IP Core. The IP Core is provided by Beckhoff Automation GmbH and different license models are offered for available FPGA devices.

- **EEPROM (ESC configuration data and application specific data)**

The EEPROM (also called Slave Information Interface, SII) contains hardware configuration information for the ESC which is loaded to the ESC's registers during power-up. The ESC activates the defined PDI so that the DPRAM can be accessed from the local host controller.

The EEPROM can be written by the configuration tool (via EtherCAT) based on the ESI file. The µC can also access the EEPROM if access rights are assigned. However, the EEPROM is always accessed via the ESC, which in turn interfaces to it via Inter-Integrated Circuit (I^2C) data bus.

- **Application Layer (AL) / Host Controller (µC)**

Application layer services, i.e. communication software and device specific software, can be implemented on a local µC. This controller then handles the following:

i. EtherCAT State Machine (ESM) in the slave device (chapter 1.3.7)

ii. Process data exchange with the slave application (e.g. application and configuration parameters, object dictionary, chapter 2.3.6)

iii. Mailbox-based protocols for acyclic data exchange (CoE, EoE, FoE, chapter 1.3.6)

iv. Optional TCP/IP stack if the device supports EoE

The µC-performance depends solely on the device application, not on the EtherCAT communication. In many cases an 8-bit µC / PIC is sufficient.

- **EtherCAT Slave Information File (ESI)**

Every EtherCAT device must be delivered with an EtherCAT Slave Information file (ESI), a device description document in XML format. Information about device functionality and settings is provided by the ESI. ESI files are used by the configuration tool to compile network information (ENI) in offline mode (e.g. process data structures, initialization commands).

Refer to the ETG.2000 EtherCAT Slave Information Specification for the description details of the ESI file. See also related description in chapter 2.4.1.

- **Individual HW/SW**

Eventually, additional device or vendor specific hardware or software is used to implement the device functionality, e.g. optics/optoelectronics in sensors, plugs in gateways, displays, etc. This hardware is connected to the host controller and is not understood here as part of the EtherCAT functionality.

1.3. EtherCAT Technology Overview.

In this chapter, basic EtherCAT slave features and functionalities are explained in a short. Refer to referenced material in chapter 1.1 for more details.

1.3.1. Frame Processing Order

The ESC provides up to 4 ports at maximum. Port 0 has to be defined as the IN-port. Slaves should provide at least two EtherCAT ports. In case the slave has two ports, ports 0 and 1 should be used (e.g. in modular devices).

Any physical EtherCAT network topology always forms a logical ring since the frame processing in a slave works like a roundabout, see Figure 4. The ESCs are connected to upstream (master) always via port 0 and to downstream (following slaves) via ports 1 to 3. The frame processing is done only once per ESC in the EtherCAT Processing Unit (EPU) which is located after port 0. Thus, returning frames will not be processed again but are only passed to the next port (as shown on port 1) or returned to port 0 (as shown on port 2).

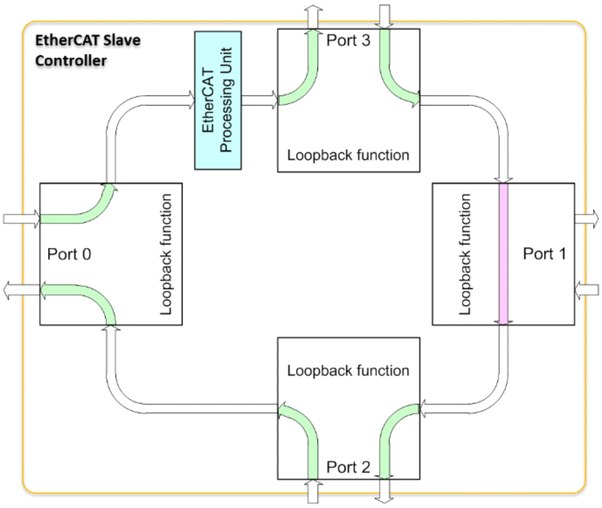


Figure 4: ESC with 4 Ports and Frame Processing Order

EtherCAT frames (Ethernet frames with EtherType 0x88A4, see Figure 5) are processed by the ESC on the fly[1]. Processing of EtherCAT datagrams is started before the complete frame has been received. Thereby Figure 5 (1) describes the EtherCAT frame as used for real-time communication. Figure 5(2) shows, that EtherCAT frames can also be routed via IP and hence, sent via sockets. However, since IP introduces jtter it is not used for realt-time communication, but might be for testing purpose. In case the frame has been corrupted, the frame checksequence (FCS) does not match and the ESC does not copy the received data to the DPRAM for the local application controller.

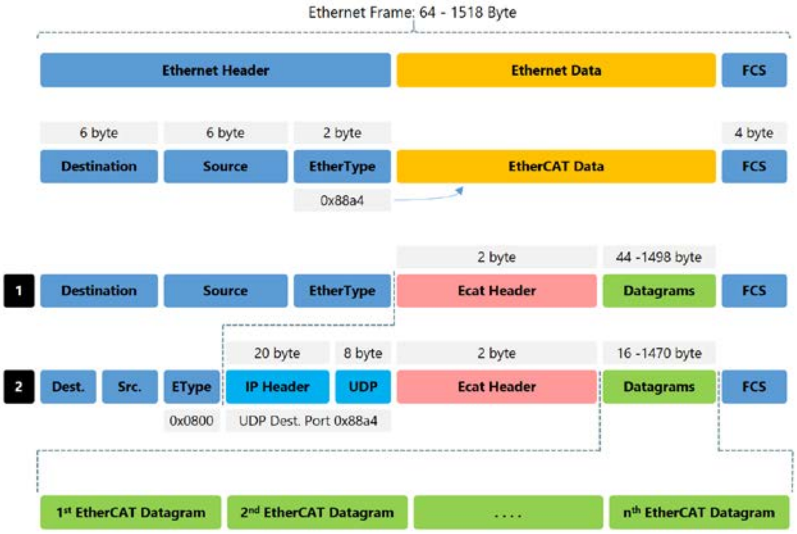


Figure 5: EtherCAT Frame Structure

1.3.2. Slave Information Interface (SII)

Since the DPRAM in the ESC is a volatile RAM, it is connected to an EEPROM (Non volatile RAM, also called Slave Information Interface, SII). The SII stores slave identity information and information about the slave's functionality corresponding to the ESI file, see Figure 6. The content of the EEPROM has to be configured by the vendor during development of the slave device. EEPROM information can be derived from the ESI file. For the SII specification, refer to ETG.1000.6 and the EtherCAT Knowledge Base.

[1] For visualization, watch https://www.youtube.com/watch?v=z2OagcHG-UU

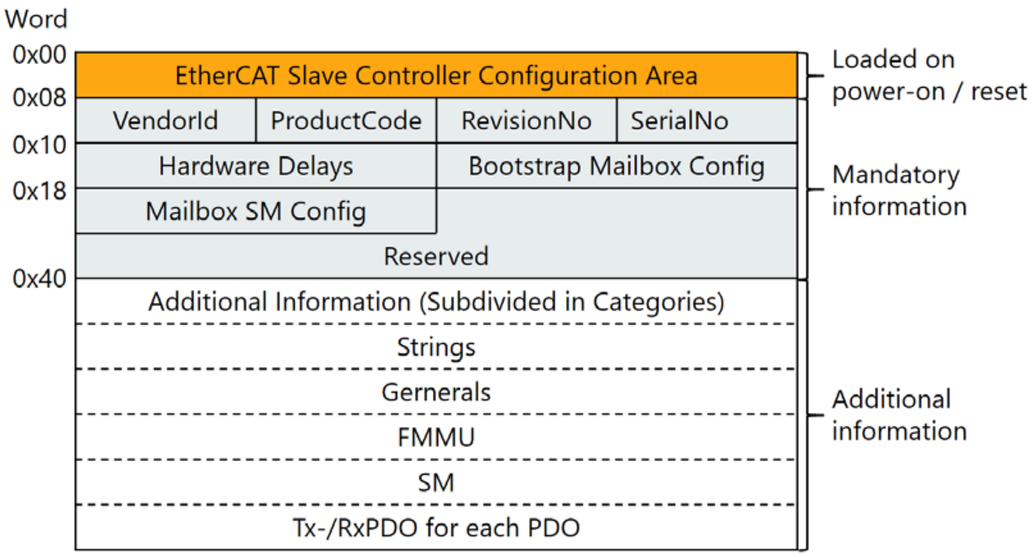


Figure 6: EEPROM Table of Register Values

1.3.3. Fieldbus Memory Management Unit (FMMU)

Fieldbus Memory Management Units are used to map data from the (logical) process data image in the master to the physical (local) memory in the slave devices. Process data in the master’s image is arranged by tasks. Related to this, the master configures via the FMMUs which EtherCAT slave devices can map data in a same EtherCAT datagram to automatically group process data. Thus, process data mapping in the master is not necessary anymore and a significant amount of CPU time and bandwidth usage are saved.

Figure 7: Mapping Example of Process Data with FMMU

1.3.4. SyncManager (SM)

Since both the EtherCAT network (master) and the PDI (local µC) access the DPRAM in the ESC, the DPRAM access needs to ensure data consistency. The SyncManager is a mechanism to protect data in the DPRAM from being accessed simultaneously. If the slave uses FMMUs, the SyncManagers for the corresponding data blocks are located between the DPRAM and the FMMU. EtherCAT SyncManagers can operate in two modes, Mailbox Mode and Buffered Mode.

Mailbox Mode

The mailbox mode implements a handshake mechanism for data exchange. EtherCAT master and µC application only get access to the buffer after the other one has finished its access. When the sender writes the buffer, the buffer is locked for writing until the receiver has read it out. The mailbox mode is typically used for application layer protocols and exchange of acyclic data (e.g. parameter settings).

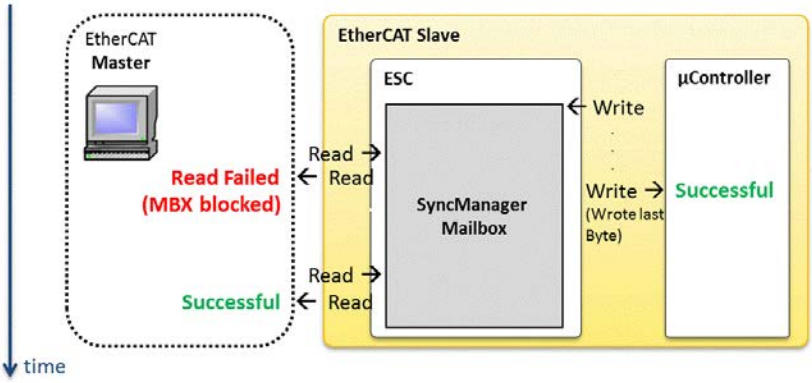


Figure 8: SyncManager in Mailbox Mode

Buffered Mode

The buffered mode is typically used for cyclic data exchange, i.e. process data since the buffered mode allows access to the communication buffer at any time for both sides, EtherCAT master and µC application. The sender can always update the content of the buffer. If the buffer is written faster than it is read out by the receiver, old data is dropped. Thus, the receiver always gets the latest consistent buffer content which was written by the sender.

Note, SyncManagers running in buffered mode need three times the process data size allocated in the DPRAM.

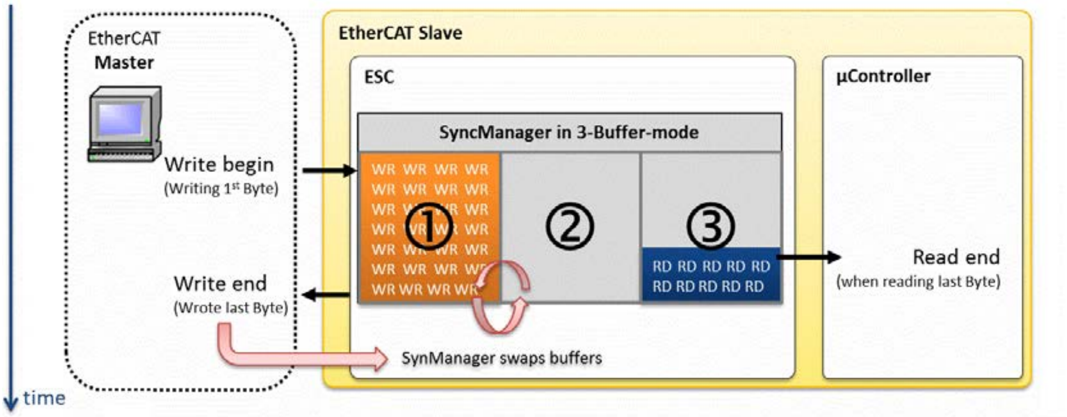

EtherCAT Master
Write begin (Writing 1st Byte)
Write end (Wrote last Byte)
EtherCAT Slave
ESC
SyncManager in 3-Buffer-mode
WR WR WR WR
①
②
③
RD RD RD RD
SynManager swaps buffers
µController
Read end (when reading last Byte)
time

Figure 9: SyncManager 3-Buffer-Mode

1.3.5. Distributed Clocks (DC)

The method of Distributed Clocks provides highly precise time synchronization between slaves in an EtherCAT network. Since DC refers to the ESC-internal clocks, synchronization time between slaves can be guaranteed to much better than 1µs.

The requirement of DC depends on the necessity of synchronization precision of the developing slave device. For instance, in machines in which multiple servo drives are functionally coupled, the axes need to be precisely synchronized to perform coherent movement. For this reason, many slaves for servo drive adopt DC in order to achieve high synchronization precision with other slaves. Thus the DC functionality should be implemented in cases of servo drive systems or I/O slaves being synchronized with servo drives.

1.3.6. Data Structure and Communication Protocols

Data is exchanged cyclically or acyclically and data sizes can be fixed or configurable. For acyclic data exchange, EtherCAT provides mailbox communication protocols (CoE, SoE, EoE, FoE, AoE). Cyclic data is exchanged in Process Data Objects (PDOs) with fixed or configurable PDO sizes. In the following, the mailbox protocols are described.

CoE: CAN application protocol over EtherCAT

This is the most commonly used EtherCAT communication protocol for acyclic data access. CoE also provides mechanisms to configure PDOs for cyclic data exchange.

Several device profiles can be applied for EtherCAT devices by using CoE. For example the drive profile CiA402 (IEC61800-7-201) is mapped to EtherCAT this way and described in more detail in the ETG.6010 Implementation Directive for the CiA402 Drive Profile.

For all other devices, the ETG.5001 Modular Device Profile Specification defines a standardized structure for the object dictionary provided by CoE. In particular, for gateways or bus couplers, these structures are enhanced by helpful configuration mechanisms.

SoE: Servo drive profile over EtherCAT

SERCOS interface™[1] is a communication interface, particularly for motion control applications. The SERCOS profile for servo drives is specified by the IEC 61800-7 standard. The mapping of this profile to EtherCAT is specified in part ETG.1000.3.

The service channel, and therefore access to all parameters and functions residing in the drive, is based on the EtherCAT mailbox. Here too, the focus is on compatibility with the existing protocol (access to value, attribute, name, units etc. of the IDNs) and expandability with regard to data length limitation. The SERCOS process data is transferred using EtherCAT slave controller mechanisms.

EoE: Ethernet over EtherCAT

The EtherCAT technology is not only fully Ethernet-compatible, but the protocol tolerates other Ethernet-based services and protocols on the same physical network. The Ethernet frames are tunneled via the EtherCAT protocol, which is the standard approach for internet applications (similar to VPN, PPPoE (DSL) etc.). The EtherCAT network is fully transparent for the Ethernet device, and the real-time characteristics are not impaired.

EtherCAT devices can additionally provide other Ethernet protocols and thus act like a standard Ethernet device. The master acts like a layer 2 switch that redirects the frames to the respective devices according to the address information. All internet technologies can therefore also be used in the EtherCAT environment: integrated web server, e-mail, FTP transfer etc.

FoE: File Access over EtherCAT

EtherCAT provides the FoE protocol for simple file access. The device e.g. runs in a boot loader state to support a firmware download to the host controller via the EtherCAT network. Standardized firmware download to devices is therefore possible, even without the support of TCP/IP.

1.3.7. EtherCAT State Machine

The slave runs a state machine to indicate which functionalities are actually available. This EtherCAT State Machine (ESM) is shown in Figure 10.

ESM requests are written by the master to the slave’s AL Control register in the ESC. If the configuration for the requested state is valid, the slave acknowledges the state by setting the AL Status register. If not, the slave sets the error flag in the AL Status register and writes an error code to the AL Status Code register.

[1] SERCOS interface is a trademark of the SERCOS International e.V.

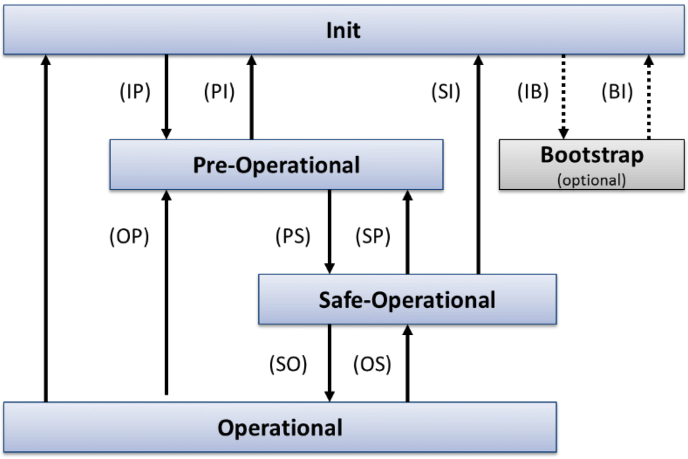


Figure 10: EtherCAT Slave State Machine

The states are described in Table 2. For further information, refer to ETG.1000.6.

Table 2: EtherCAT State Machine Description

State	Available Functions
Init (INIT)	Init state. No communication on the application layer is available. The master has access only to the DL-information registers.
Pre-Operational (PREOP)	Pre-Operational state. Mailbox communication on the application layer available, but no process data communication available.
Safe-Operational (SAFEOP)	Safe-Operational state. Mailbox communication on the application layer, process (input) data communication available. In SafeOp only inputs are evaluated; outputs are kept in 'safe' state.
Operational (OP)	Operational state. Process data inputs and outputs are valid.
Bootstrap (BOOT)	Bootstrap state. Optional but recommended if firmware updates necessary No process data communication. Communication only via mailbox on Application Layer available. Special mailbox configuration is possible, e.g. larger mailbox size. In this state usually the FoE protocol is used for firmware download.

The initialization information for every EtherCAT state transition is based on the ESI, a network configurator saves it to the EtherCAT network information file (ENI). Each slave receives its required initialization commands for each state transition. The EtherCAT master maintains independent state machines per EtherCAT slave in the network. The state transition control sequences are shown in Figure 11.

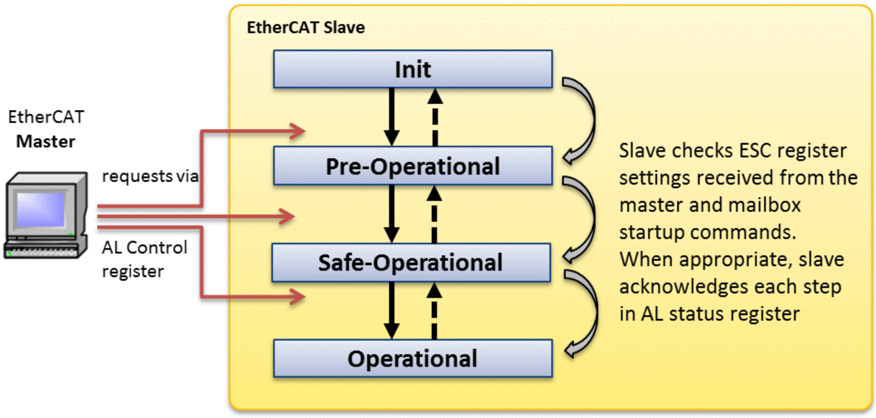

EtherCAT Slave
Init
Pre-Operational
Safe-Operational
Operational
EtherCAT
Master
requests via
AL Control
register
Slave checks ESC register settings received from the master and mailbox startup commands.
When appropriate, slave acknowledges each step in AL status register

Figure 11: EtherCAT Network Initialization

For the development of (complex, i.e. slaves using a host controller) EtherCAT slaves, the handling of the state transition commands is mandatory. The prerequisite for the state machine functionality is the successful reception and acknowledgement of the state transition requests in the EtherCAT slave device (reading/writing AL, control/AL status registers). When the master sends a state request, the acknowledgement must not be given before the register and application configuration corresponding to the requested state is validated by the local µC. Full data exchange with the master is enabled when the slave switches to the operational state. The state machine handling is subject to tests in the EtherCAT Conformance Test Tool.

Table 3: EtherCAT State Machine Transitions[1]

Transition	Master to Slave Settings Description
IP	Master reads VendorID, ProductCode and RevisionNumber from EEPROM, and configures DL control registers (register 0x0100:0x0103) SyncManager registers (registers 0x800+) for mailbox communication, Initialization for DC clock synchronization (if supported). Master requests PreOp state by writing the AL Control register (register 0x120) and waits for status confirmation via the AL Status register (register 0x130).
PS	Master configures parameters using mailbox communication, i.e. Process Data Mapping if flexible, registers for process data SyncManagers, FMMU registers (0x600 and following). Master requests SafeOp state (AL Control register 0x0130[2]) and waits for confirmation via AL Status register.
SO	Master sends valid Outputs and requests Op state (AL Control register 0x0130[4], confirmation in AL Status register)
Error Init Error PreOp Error SafeOp	Incorrect ESC register configuration (DC, FMMU, SyncManager, etc.). The AL Status Code register (register 0x134) indicates error reasons.

[1] Detailed description is available in the ETG.1000 EtherCAT Communication Specification (Part 6, Table 103).

2. EtherCAT Slave Implementation

This chapter shows the procedure for a typical EtherCAT slave implementation process. The overview to the steps is given in chapter 2.1. The steps are described in more detail in the denoted chapters. Chapter 2.2 contains details for administrative organization. Chapters 2.3 to 2.6 contain the detailed descriptions of the development steps. Herein, some application notes are given as well. Chapters 2.7 and 3 describe support which is done by the ETG.

2.1. General Procedure – Step by Step

A well proven approach to an EtherCAT slave implementation is given in the following figure.

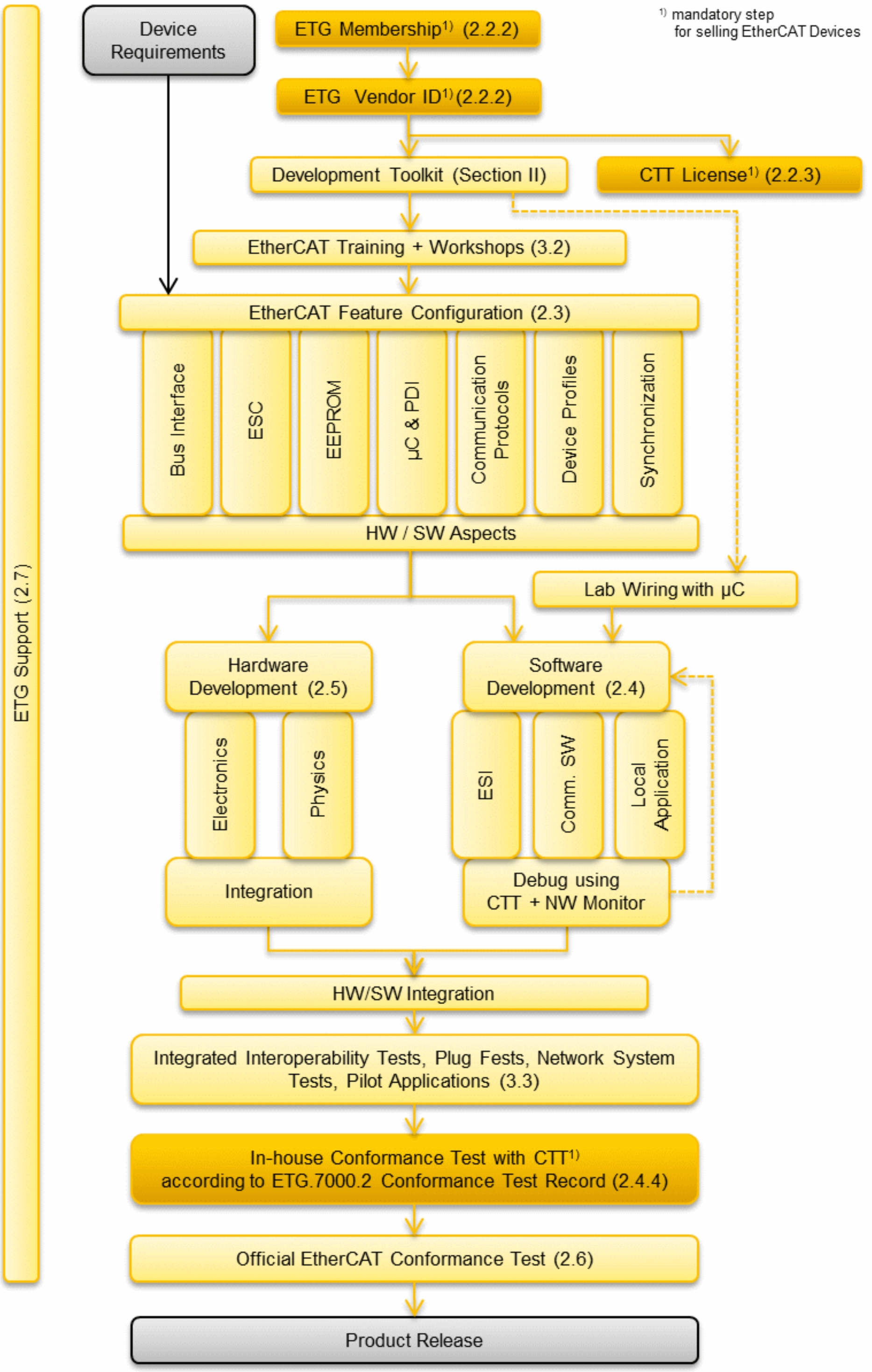

1) mandatory step
for selling EtherCAT Devices
ETG Support (2.7)
Device Requirements
ETG Membership1) (2.2.2)
ETG Vendor ID1) (2.2.2)
Development Toolkit (Section II)
CTT License1) (2.2.3)
EtherCAT Training + Workshops (3.2)
EtherCAT Feature Configuration (2.3)
Bus Interface
ESC
EEPROM
µC & PDI
Communication Protocols
Device Profiles
Synchronization
HW / SW Aspects
Lab Wiring with µC
Hardware Development (2.5)
Electronics
Physics
Integration
Software Development (2.4)
ESI
Comm. SW
Local Application
Debug using CTT + NW Monitor
HW/SW Integration
Integrated Interoperability Tests, Plug Fests, Network System Tests, Pilot Applications (3.3)
In-house Conformance Test with CTT1) according to ETG.7000.2 Conformance Test Record (2.4.4)
Official EtherCAT Conformance Test (2.6)
Product Release

Figure 12: EtherCAT Device Development Procedure

2.2. Administrative Organization

2.2.1. Development Time

To develop a new running slave system, operated by a standard EtherCAT master, about 6-8 weeks are feasible to get a working solution. Herein, parts of the own application development are already included.

The hardware design of the device depends on device type (with or without µC) and the amount and type of ports (MII or LVDS). Table 4 shows the components needed for a slave device.

Table 4: Components to develop/configure for EtherCAT Devices

	Category	Simple Device (no µC, dig. I/O)	Complex Device (with µC)
Hardware	**Host controller**	--	Microcontroller Programmable Memory (ROM) RUN (ev. ERR) LED
	ESC	ESC (ASIC/IP Core) EEPROM	
	Port connection	MII: Plug, TRAFO, PHY, R/C Link/Activity LEDs LVDS: Capacitor-/Resistor combinations (R/C) (Link/Activity LEDs)	
	Device casing	Coverage design, ev. additional individual hardware etc.	
Software	**Host application**	--	Microcontroller local application/FW EtherCAT communication
	Device description	ESI file EEPPROM configuration	
	Documentation	EtherCAT slave device documentation	

2.2.2. ETG Membership and Vendor ID

Each EtherCAT compliant device has to carry a worldwide unique Vendor ID assigned by the EtherCAT Technology Group (s. chapter 3), which requires ETG membership as well.

ETG membership is free of charge and covered by the ETG Membership By-Laws. For application send your membership request in an email to info@ethercat.org.

The Vendor ID usage is covered by the ETG.9002 Vendor ID Policy. The application for the Vendor ID can be done online (membership login data is required). The Vendor ID is free of charge as well. The EtherCAT Vendor ID is mandatory to meet the Conformance Test requirements.

2.2.3. EtherCAT Conformance Test Tool License

There are two reasons why to buy an EtherCAT Conformance Test Tool (CTT) license.

- The CTT assists EtherCAT device development by checking protocol compliance in-house and supports preparation for the official EtherCAT Conformance Test (chapter 3.4). It also delivers a good deal of development and testing support by many built-in feature, including a remote control interface to run it by a script.
- The application of the CTT for in-house tests is mandatory when selling the device to the market.

The tests performed by the CTT are specified by the ETG Working Group Conformance. The CTT software is provided by Beckhoff Automation GmbH & Co. KG.

Important to know: To guarantee long-time availability of the CTT, i.e. ensuring maintenance of the software such as adding support for new operating systems, the ETG membership assembly 2008 decided unanimously for the following model: The CTT comes on a subscription basis extending itself automatically every year. A new license file is automatically being provided to the vendors. Each EtherCAT slave manufacturer who offers EtherCAT slaves to the market or builds own slaves to integrate them into their machines, shall obtain and maintain a valid subscription.

Before canceling the subscription, checking if this would violate the policies (especially the Conformance Test Policy, ETG.9003) is obligatory. In case of a cancellation, usually a standard lead-time of 3 month before the renewal of the license applies.

Support is provided by ETG (conformance@ethercat.org).

2.3. EtherCAT Slave Design

EtherCAT features are to be selected according to the device requirements. Thus, to develop an EtherCAT slave device, the developer should be conscious about the requirements of the device to decide which characteristic is to be chosen for every EtherCAT feature.

In the following, an overview to the design criteria is given of which the ESC is the most important EtherCAT characteristic. The configuration of these criteria is finally stored in the ESI file and the EEPROM.

2.3.1. Bus Interface to EtherCAT Network

Support of the desired bus interface(s) must be regarded in the selection of the ESC. It is one of the main criteria for ESC types.

For stand-alone devices which are connected to the network via 100BaseTX or 100BaseFX cable, Media Independent Interface (MII) is used. For modular devices which are connected via a backbone connection, LVDS (Low Voltage Differential Signaling) is available as internal physical layer. To access external interfaces from modular devices a converter from 100Base technology to LVDS physics is necessary.

Application Note: A stand-alone device should support at least two MII ports (RJ45 or M12 D-Code connectors) to provide line connection. The logical port for connection is determined based on the number of ports being used. For standard 2 port usage, port0 and port1 are used. The PHYs should be selected according to the PHY Selection Guide.

2.3.2. EtherCAT Slave Controller (ESC) and PDI

The ESC is the controller which provides the communication interface between the EtherCAT network and the host controller (device application controller) or the digital I/O (if no host controller is used).

Basically, the ESC can be implemented as ASIC or as FPGA with IP Core. The EtherCAT functionality is the same for both types, so the choice which type to use is up to the vendor. If preferring an ASIC, an additional EEPROM is necessary and the DPRAM may be limited to less than 64kbyte (depending on the ESC).

If know-how of FPGA programming is available and intellectual property (IP core) is already at hand, the choice for an FPGA implementation is obvious and the IP Core only needs to be adapted to the EtherCAT communication. An FPGA may also be an option if hardware space for both an ASIC and an EEPROM is not available.

An overview of available ASICS and FPGAs is given by the ETG in chapter **Error! Reference source not found.** of section II or in the ESC Overview. In the following, the ESC selection criteria are discussed in more detail.

- **Number and type of EtherCAT ports (MII, LVDS)**

Basically, EtherCAT devices have two ports so that they can be connected in a line topology. The number of ports and port type are key selection criteria of ESCs.

- **Interface for process data exchange (PDI)**

For ASICs, simple devices usually require no application logic in software (µC) but only digital I/O. Complex devices operate via a serial peripheral interface (SPI) or 8/16 bit synchronous or asynchronous microcontroller interface (MCI) via parallel port.

If using an EtherCAT IP core, the FPGA specific on-board-bus is applied as PDI since ESC, EEPROM and µC are integrated in the IP Core. For an example on Altera devices Avalon is used resp. OPB on Xilinx devices.

- **DPRAM size and number of SyncManagers**

The DPRAM is used for exchange of cyclic and acyclic data via the EtherCAT network. SyncManagers ensure data consistency within the DPRAM. Each ESC has 4kByte of *registers* (addresses 0x0000 to 0x0FFF) which are reserved for (EtherCAT and PDI communication) configuration settings.

Mailbox and process data is exchanged via additional *DPRAM* (also called user memory). EtherCAT allows addressing of user memory of up to 60kBytes. ASICs provide between 1kByte and 8kByte of DPRAM, IP Cores can be configured to provide the full 60kByte of user memory.

Application Note: The standard SyncManager (SM) configuration is

- 1 SM per acyclic data output (mailbox out, master to slave)
- 1 SM for acyclic data input (mailbox in, slave to master)
- 1 SM for cyclic data output (process data out, master to slave)
- 1 SM for cyclic data input (process data in, slave to master)

For process data, SM running in 3-buffer-mode need three times the length of actual process data for physical memory. The following table shows a schema of how to allocate the length for the 4 SM.

Table 5: DPRAM Size Calculation Example

	SyncManager	Buffer Count	Length [Byte]	Total length [Byte]
SM0	Output Mailbox	1	L_MbxOut	1*L_MbxOut
SM1	Input Mailbox	1	L_MbxIn	+ 1*L_MbxIn
SM2	Outputs	3	L_Out (TxPDO)	+ 3*L_Out
SM3	Inputs	3	L_In (RxPDO)	+ 3*L_In
				∑ DPRAM size

SyncManagers are enabled by the following settings of the master during network initialization.

- Physical address of ESC
- Data length
- SyncManager control input:

i. Operation mode (mailbox-mode/3-buffer-mode)

v. Access direction (Read direction/Write direction)

vi. Interrupt settings (Valid/Invalid)

vii. SyncManager watchdog setting (Valid/Invalid)

viii. SyncManager setting (Valid/Invalid)

The default values are set in the ESI (chapter 2.4.1); the master initializes the SyncManager using the values from the ESI.

- **Number of Fieldbus Memory Management Units (FMMUs)**

In an EtherCAT network, the memory of all slaves can be compiled in the master to a logical memory. This logical memory is managed by FMMUs to map logical addresses to physical addresses in the slaves. For the FMMU configuration in a device, each consistent output and each consistent input block needs one FMMU and an additional FMMU for mailbox status response is necessary.

Application Note: The standard configuration is one FMMU per each, cyclic output and cyclic input data block, optionally an additional one for mapping the mailbox response availability flag into process data (thus, no polling of mailboxes is necessary). If the outputs and inputs are grouped e.g. like in Table 5, 3 FMMUs are configured, see Table 6.

Table 6: FMMU Configuration

FMMU	Assigned SyncManager	Name	Length [Byte]
1	SM2	Outputs	L_Out (TxPDO)
2	SM3	Inputs	L_In (RxPDO)
3	SM0 & SM1	Mbx-SM Status Flags	Mbx In/Out Length

- **Distributed Clocks (DCs) for synchronization with other slave devices**

Evaluate if the device should support high precise synchronization with other slave devices. If so, DCs should be supported by the selected ESC. Distributed Clocks refer to the DC function for EtherCAT slaves (chapter 1.3.5). The times held by slaves are adjusted with this mechanism and thus enable precise synchronization of the nodes in the EtherCAT network.

2.3.3. EEPROM

The EEPROM is mounted outside the ESC and connected via I^{2}C with point-to-point link. According to the size of the EEPROM the EEPROM_SIZE signal should be set. For more details, refer to the Knowledge Base, chapter “EEPROM".

For EEPROM (SII) Enhanced Link Detection setting, refer to documentation of the ESC vendor.

2.3.4. Application Controller (Host Controller, µC)

If a local software application provides the device functionality, any 8 or 16 bit synchronous or asynchronous microcontroller can be connected to the ESC. The application controller communicates with the ESC via the Process Data Interfaces (PDI).

To adapt the application software on the host controller to the ESC, sample software stacks are available for communication implementation, e.g. the Slave Stack Code (SSC). If the device is a 32 bit digital I/O interface, no application controller or additional communication software is necessary.

In most cases, manufacturers can use a familiar microcontroller type as application controller in the EtherCAT device. If application software already exists, e.g. for a different fieldbus, it can be used for the EtherCAT device as well.

The source code for communications software on the host controller allocates about 70kByte. The following features are a typical configuration (referring to the Slave Stack Code):

- EtherCAT State Machine (ESM), including error handling
- Device diagnosis
- Master-Slave data synchronization with SyncManager event (no DCs)
- Mailbox CoE
- Object Dictionary (20 objects) for process data objects
- CoE services, including CoE Info services, no segmented transfer

2.3.5. Application Layer Communication Protocols

In EtherCAT, several protocols are available (see chapter 1.3.6) for the application layer to implement the required specification of the product development. When to apply them is described here.

- CAN application protocol over EtherCAT (CoE)
  To provide acyclic data exchange as well as mechanisms to configure PDOs for cyclic data exchange in a structured way, CoE (with SDO-Info support) should be implemented.

- Servo drive profile over EtherCAT (SoE)
  SoE is an alternative drive profile to the CiA402 drive profile. It is often used by drive manufacturers which are familiar with the SERCOS interface.

- Ethernet over EtherCAT (EoE)
  EoE is usually used to provide webserver interfaces via EtherCAT. It is also used for devices providing decentral standard Ethernet ports.

- File Access over EtherCAT (FoE)
  If the device should support firmware download via EtherCAT, FoE should be supported. FoE is based on TFTP. It provides fast file transfer and small protocol implementation.

- ADS over EtherCAT (AoE)
  AoE is specified to be used for use cases as they appear e.g. for fieldbus gateways: fieldbus slaves behind the gateway, including their type of object dictionary can be accessed using AoE. It is routable and allows parallel requests to fieldbus slaves behind the gateway. AoE does not provide a semantic concept (data types, structure, etc.) as CoE does.

Application Note: An exemplary CoE implementation is shown below.

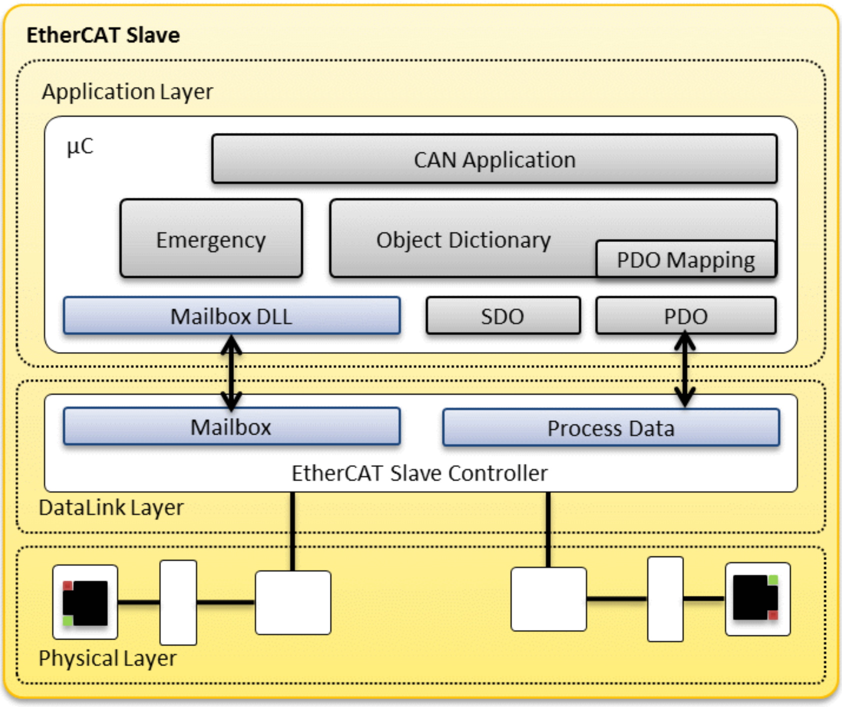


Figure 13: ESC Structure for CAN application profile Applications

The user application runs the device specific software on the µC to implement device features. Slave source codes offered by EtherCAT stack vendors can be found on the product section (text filter: Slave Stack) and used to develop this application or to adapt existing software to EtherCAT.

Application Note: EtherCAT Slave Stack Code (SSC).

The SSC is a free sample code from Beckhoff which provides an interface to the ESC. For hardware independent software development, the SSC runs on several evaluation kits and can be customized for implementation in accordance with the product specification. Figure 14 shows the SSC structure with the interfaces to the user specific device application and the ESC.

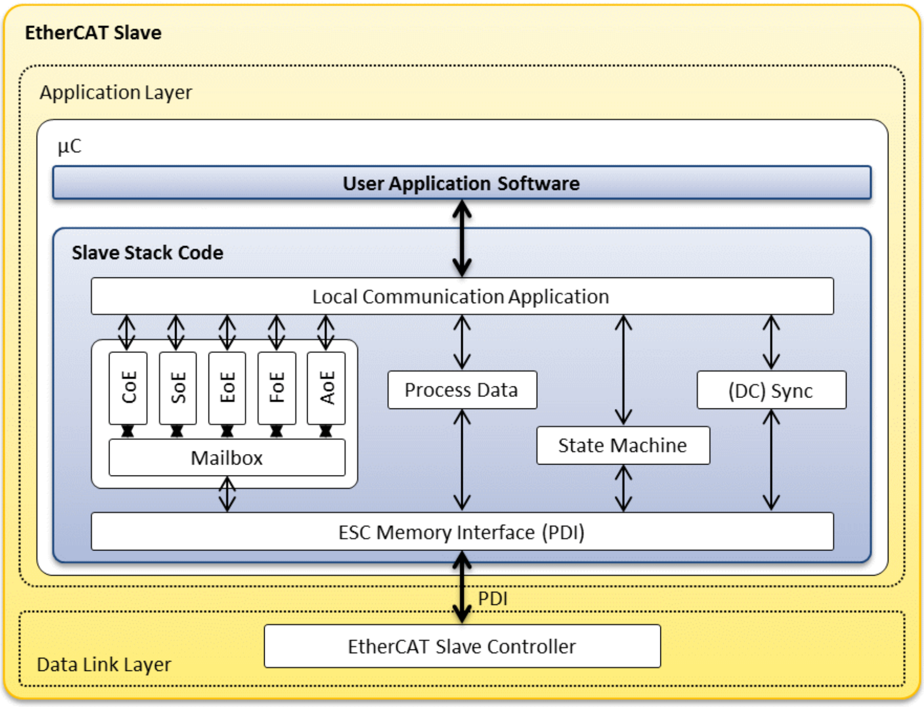


Figure 14: Slave Stack Code Overview

Application Note: EtherCAT Slave Protocol Stack.

Hilscher or HMS offer a Slave Control Stack based on its netX or AnyBus hardware with Dual Port Memory interface (DPM) and it is available for the user application with an API. Figure 15 shows the protocol stack architecture with interfaces to the ESC and the user application.

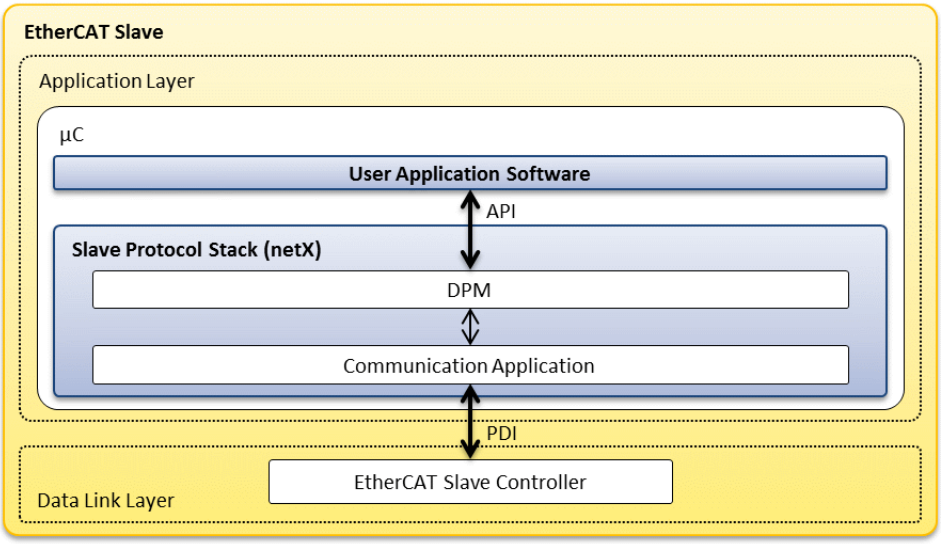
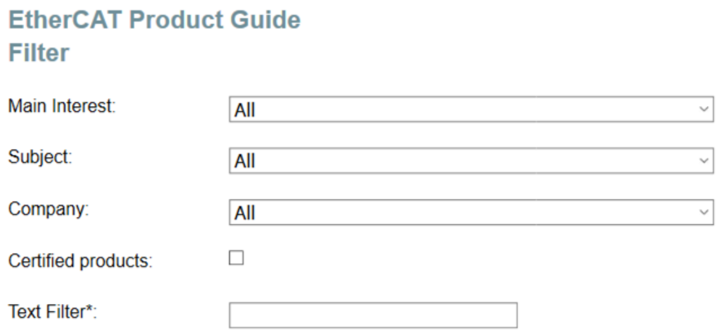

EtherCAT Slave
Application Layer
µC
User Application Software
API
Slave Protocol Stack (netX)
DPM
Communication Application
PDI
Data Link Layer
EtherCAT Slave Controller

Figure 15: Slave Control Stack

A list of other available sample stacks can be obtained on the official EtherCAT Product Section of the ETG website with the following filter options.



EtherCAT Product Guide
Filter

Main Interest: All

Subject: All

Company: All

Certified products: □

Text Filter*:

Figure 16: product filter SSC

2.3.6. Device Profiles

Device profiles define a common application interface for specific devices. Drive profiles define the identifier (CoE Index) and data type for the status word and control word, for set point and current value, and standard parameters. The master application such as a PLC program, can then use the same data structure for drives of different vendors.

The EtherCAT specifications define different profiles. Some examples are:

- Drive Profile
  The drive profiles are according to IEC 61800-7 and include the mapping of the CiA406 Drive Profile to EtherCAT and the Sercos™ Drive Profile to EtherCAT. The ETG.6010 Implementation Directive describes more details and helps understanding the CiA406 Drive Profile on EtherCAT.

- Modular Device Profile (MDP)
  The MDP provides general rules for the structuring of the CoE object dictionary. Device specific profiles are available, too, such as for fieldbus gateways. It is specified in ETG.5001.

- Semi Device Profile
  It defines a series of profiles for different devices used in the semiconductor manufacturing industry, mainly focusing on the process part, such as pumps, valves, mass flow controllers, temperature controllers. It is specified in ETG.5003 and its basic structure is very close to ETG.5001.

The object dictionary can be described as a two dimensional list. Each list entry is identified by an index (0x0000 – 0xFFFF) which represents an object. Each object can contain up to 255 subindices, also called object entries. The object list is structured in different areas, see Table 7.

Table 7: The Modular Device Profile Object Dictionary

Object Index Range	Reserved for		Comment
0x0000 – 0x0FFF	Data Type Area		Protected registers for ESC configuration
0x1000 – 0x1FFF	Communication Area		Communication parameters, settings, etc.
0x2000 – 0x5FFF	Manufacturer specific Area		
0x6000 – 0x6FFF	Profile Specific Area	Input Area	Process data input objects (mapped to TxPDOs)
0x7000 – 0x7FFF		Output Area	Process data output objects (mapped to RxPDOs)
0x8000 – 0x8FFF		Configuration Area	Process data configuration and settings objects
0x9000 – 0x9FFF		Information Area	Scanned information from modules
0xA000 – 0xAFFF		Diagnosis Area	Diagnostic, status, statistic or other information
0xB000 – 0xBFFF		Service Transfer Area	Service objects
0xC000 – 0xEFFF		Reserved Area	
0xF000 – 0xFFFF		Device Area	Parameters belonging to the device

The idea of the MDP is to provide a basic structure for masters and configuration tools to handle slaves with complex (modular) structure easily. The user has the advantage, that if the slave’s variables are sorted in an MDP style, so he can find the different data types by identical patterns.

The MDP can be applied to various types of devices. It is applicable to multiple axis servo drive system of various functionality groups, such as positioning, torque and velocity control. It is further applicable to gateway between different fieldbuses, i.e., Profibus, DeviceNet. Modular devices are driven by two aspects:

- **Comprise physically connectable modules and plurality of functionalities.**

The MDP imagines slaves which consist of one or several modules. A module can be hardware which is connected/disconnected to a slave. Examples are gateways between EtherCAT and e.g. CANopen or a bus coupler between EtherCAT and a proprietary backbone bus.

- **Comprise plurality of channels directly being connected to the EtherCAT network.**

A module can also be a logical module which describes data sets, e.g. a drive which supports a velocity controlled mode and a position controlled mode – the MDP would describe the data as two modules, one for each mode.

No matter what kind of module is described it needs more or less the same information categories, which are organized in the profile specific index range (Table 7).

Application Note: Modular Device Profile Structure.

Consider an MDP for a line of slave device modules which are connected together on a backbone layer via LVDS and via a coupler with MII. Figure 17 shows a schema how to define device profiles such that a modular profile dictionary is set up for the slave device line.

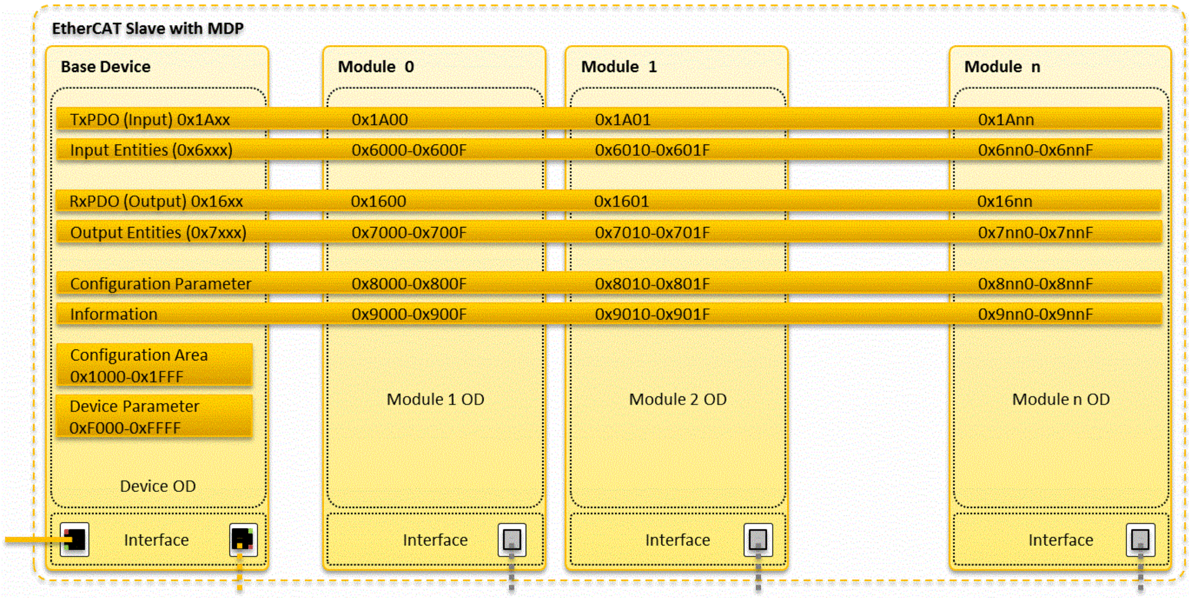


Figure 17: MDP Schema for Modular Devices

2.3.7. Synchronization among Slaves and the Master

EtherCAT provides various synchronization options. There are three different types of synchronization methods available.

- **Freerun**

The slave device application runs independently of the EtherCAT cycle and is triggered by a local timer in the ESC.

- **Synchronous with frame reception (Synchronization with SyncManager event)**

The slave device application is triggered when new process data is received. The synchronization accuracy depends on the jitter of the message reception and the delay between the other network nodes.

- **Distributed Clocks (DC, Synchronization with SYNC0/SYNC1 event)**

The ESCs contain a nanosecond based timer (DC timer) to provide precise synchronization and time stamping. The slave device application is triggered with an additional interrupt signal, which is based on the DC time and is produced by the ESC. Every DC timer in the network is aligned to a reference DC clock.

Application Note: The ESC system time is stored in a 64 bit value. This data size allows representation of more than 500 years. The latter 32 bits represent approximately 4.2 seconds. Refer to the datasheet of the specific ESC for details since some ESC use 32 bit length.

Initial value: 00:00:00 January 1, 2001
Unit: 1ns

Definition of a Reference Clock

One EtherCAT slave (which usually is the first slave that uses DC) is determined as the reference clock and becomes the clock base for the master as well as for other DC slaves. The reference clock is periodically provided to other slaves. The reference clock is adjustable by an external "global reference clock".

Function and Operation of DC

The slave synchronization is established during initialization of the ENI in the master. With EtherCAT, the 3 DC time synchronization functions enable highly accurate timing synchronization.

- **Measurement/Calculation of the propagation delay time**

  During initialization procedure of the network, the master calculates the propagation delay, including the delay caused by cables and ESC, and sets the delay as slave delay. The delay calculation algorithm is basically defined the ETG.1000.4 and further described e.g. in the ET1100 Datasheet (section I, chapter 9.1.2). After establishment of the slave DC, the EtherCAT master periodically sends Auto increment Read Multiple Write (ARMW) commands to read the time from the reference clock and write it to all other DC slaves.

- **Drift compensation**

  The master periodically reads out the time from the reference clock using the ARMW and writes the time to the other DC slaves. The deviation of time data held by the slave is thus minimized.

- **Offset compensation:**

  Offset compensation refers to function of adjusting the system time (e.g. the calendar time) held by the EtherCAT master and the time held by slave. The slave can be synchronized by the EtherCAT master by writing into the slave the deviation of time between the system time of the master and the reference clock.

Interrupt signal

After establishment of DC by the master, the ESC generates fixed time interrupt signals to the PDI, i.e. the µC. Thus, the slave is able to create a constant period. There are following 3 types of generation of interrupt signals.

- SYNC/LATCH0
- SYNC/LATCH1
- IRQ (Interrupt occurs by generation of SYNC0/SYNC1and mask register setting)

Note that the SYNC0/SYNC1 interrupt signals cannot be used when using the ESC LATCH0/LATCH1 function. This restriction is due to SYNC/LATCH signal lines being a shared pin.

The latch function is a function which maintains time stamp in response to latch signal input on the ESC, and activate/deactivate timing edges can be set.

2.3.8. Firmware Update

The EtherCAT specifications defines the FoE mailbox protocol for firmware update in Bootstrap mode of the EtherCAT State Machine. In addition, it defines FoE Error Codes and AL Status Codes which can be used to report certain errors which may occur during a firmware update procedure. However, there is no more specific description of the firmware update process as in the *Semi Device Profile Specification* of ETG.5003 the part 2 “Firmware Update Specification” specified. EtherCAT devices supporting the “Semi Device Profile” (CoE Object 0x1000 = 5003dec) must also support the firmware update mechanisms defined in part 2. Even though it is not mandatory for any other EtherCAT slave device to do so, this part provides a good guideline for a firmware update implementation on any EtherCAT slave. Some details covered by this description are:

- Slave accessibility in case of failed FW update
- ESC reset behavior
- Device documentation
- EEPROM update
- FW version and functionality verification

Therefore, it is recommended to use ETG.5003-2 as a guideline for firmware update implementation.

2.4. Tools for EtherCAT Slave Development

Table 8 lists tools that may be useful for EtherCAT device development. Some tools are described in more detail with their application purpose in the following subsections.

Note the Conformance Test Tool is mandatory for slave device vendors.

Table 8: Tools

	Tool	Description and Access
Network Configuration	EtherCAT Configurator	Configurator for loading XML device descriptions (ESI) and for generating XML network configuration descriptions (ENI). Several EtherCAT Masters already include an EtherCAT Configuration Tool. Visit the official EtherCAT Product Section of the ETG website for the variety of configuration tools. Main Interest: Development Systems, Tools Subject: Configuration Tools For development purposes, an EtherCAT Configuration Tool with master (e.g. TwinCAT) can be freely downloaded from the website of the master provider
Development	XML Editor	Used to edit or view EtherCAT Slave Information (ESI) files. Any browser or text editor can be used, as well as the CTT. Further Tools: Altova XML Spy (extensive xml editor, license fee required) Peter's XML editor (freeware) XML Notepad (freeware)
	Hex File Editor	Used to convert bitmap images (vendor or device logos) to a hex value which is needed in the ESI. Any hex editor is fine, here are two examples: HxD (freeware) Mirkes TinyHexer (freeware)
Diagnosis	Network Monitor	Wireshark (former Ethereal) can be used to monitor frame communication of EtherCAT networks. Wireshark is freeware and has already included a parser for comfortable EtherCAT frame analysis. Available for Linux and Windows
	EtherCAT Conformance Test Tool (CTT)	The Conformance Test Tool is used to check EtherCAT protocol compliance in-house. The test tool is provided by Beckhoff Automation GmbH & Co. KG. Please contact ctt@beckhoff.com
	Further Tools	Also consult the official EtherCAT Product Section of the ETG website for a continuative list of tools.

2.4.1. XML Editor for Generating ESI files

The vendor needs to deliver the device with an ESI file, since when designing an EtherCAT network, the user requires to generate the ENI file using a configuration tool and the ESI files of the slaves. Slave specific information (manufacturer, product information, profile, object, process data, sync or non-sync, sync manager setting) is registered to the ESI file in XML format. A single ESI file may include multiple slave devices' information.

The ESI file is defined with the ETG.2000 EtherCAT Slave Information specification. The structure of an ESI file is defined in the EtherCATInfo.xsd XML schema document, see Figure 18. By applying the XML schema to an XML editor, syntax checks can be made on the ESI description to avoid basic errors. The XML schema as well as a sample ESI file is available from ETG.2001 EtherCAT Slave Information Annotations.

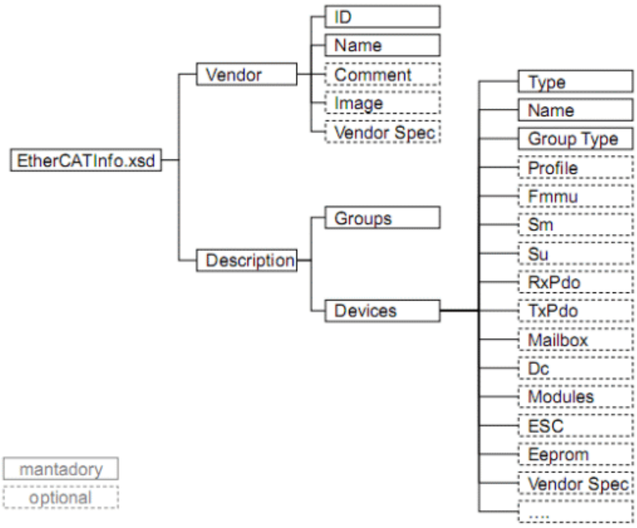


Figure 18: ESI Structure (EtherCATInfo.xsd)

A text editor or (graphical) XML editor software may be used to edit the ESI file. The CTT also provides an editing environment for ESI files as shown below.

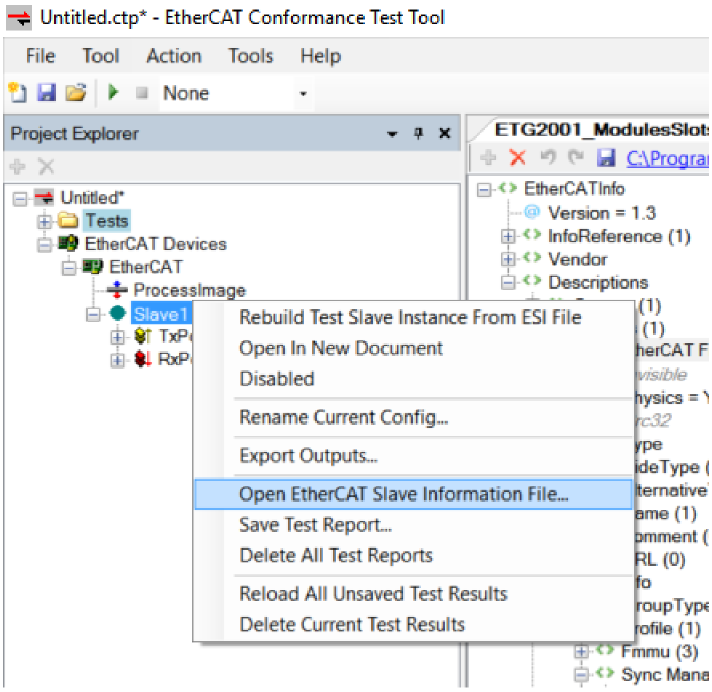


Figure 19: ESI File editing using CTT

Any other popular editor software can also be used for XML editing but for those who are looking for one, the example below may be a field-proven option.

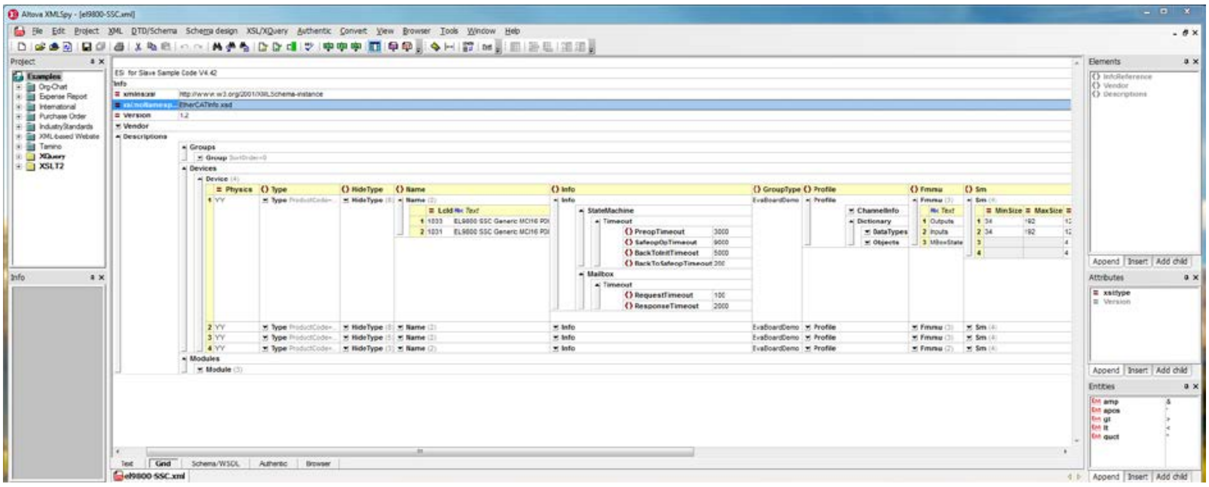
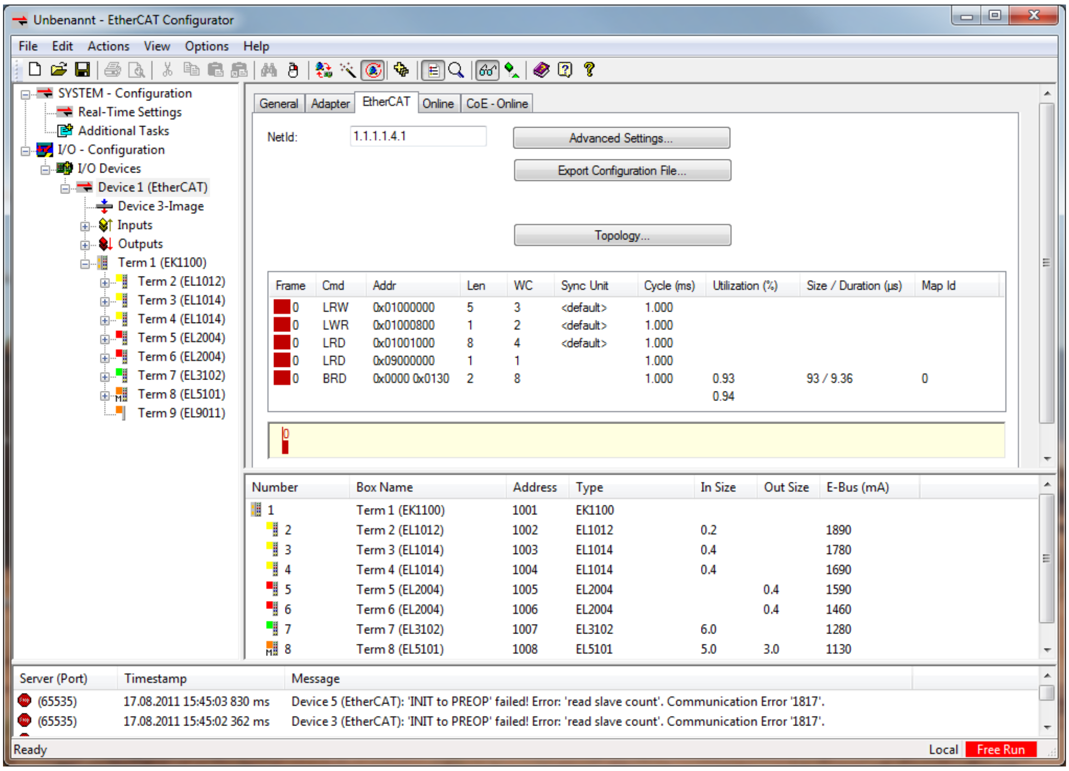

Altova XMLSpy - [el9800-SSC.xml]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic Convert View Browser Tools Window Help
Project
Examples
Info
Elements
Attributes
Entities
Text Grid Schema/WSDL Authentic Browser
el9800-SSC.xml

Figure 20: ESI File Generation using a Graphical Editor (Altova XML Spy®)

2.4.2. EtherCAT Network Configurator and Master Software

For EtherCAT network configuration, an EtherCAT Network Configurator is necessary which loads ESI files and generates an ENI file. Available software can be found on the product section of the ETG website. For example, TwinCAT has a built in network configurator and is also available as 7-day trial software.



NetId: 1.1.1.1.4.1

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
🟥 0	LRW	0x01000000	5	3	<default>	1.000			
🟥 0	LWR	0x01000800	1	2	<default>	1.000			
🟥 0	LRD	0x01001000	8	4	<default>	1.000			
🟥 0	LRD	0x09000000	1	1		1.000			
🟥 0	BRD	0x0000 0x0130	2	8		1.000	0.93 0.94	93 / 9.36	0

Number	Box Name	Address	Type	In Size	Out Size	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL1012)	1002	EL1012	0.2		1890
3	Term 3 (EL1014)	1003	EL1014	0.4		1780
4	Term 4 (EL1014)	1004	EL1014	0.4		1690
5	Term 5 (EL2004)	1005	EL2004		0.4	1590
6	Term 6 (EL2004)	1006	EL2004		0.4	1460
7	Term 7 (EL3102)	1007	EL3102	6.0		1280
8	Term 8 (EL5101)	1008	EL5101	5.0	3.0	1130

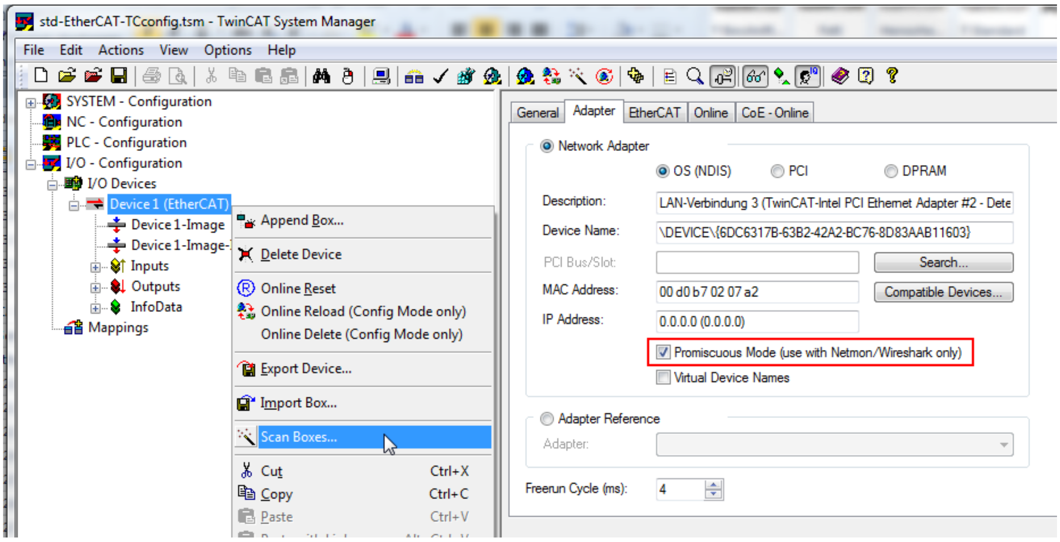
Server (Port)	Timestamp	Message
(65535)	17.08.2011 15:45:03 830 ms	Device 5 (EtherCAT): 'INIT to PREOP' failed! Error: 'read slave count'. Communication Error '1817'.
(65535)	17.08.2011 15:45:02 362 ms	Device 3 (EtherCAT): 'INIT to PREOP' failed! Error: 'read slave count'. Communication Error '1817'.

Figure 21: EtherCAT Network Configurator

Software for a master becomes necessary when running an EtherCAT network or debugging a slave device. The ESI file of the developing slave device needs to be stored in the masters EtherCAT device repository. To set up a small EtherCAT network with a master and a slave device, refer to chapter 1.2.

A list of available masters can be found on the product section (text filter: master) of the ETG website. For example, TwinCAT from the Beckhoff Automation is available as trial version. In TwinCAT System

Manager, right click on I/O Device, scan devices and further scan for boxes. Refer to the TwinCAT manual for the subsequent steps to assemble an EtherCAT network.



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

Description: LAN-Verbindung 3 (TwinCAT-Intel PCI Ethernet Adapter #2 - Dete

Device Name: \DEVICE\{6DC6317B-63B2-42A2-BC76-8D83AAB11603}

PCI Bus/Slot:

MAC Address: 00 d0 b7 02 07 a2

IP Address: 0.0.0.0 (0.0.0.0)

☑ Promiscuous Mode (use with Netmon/Wireshark only)

□ Virtual Device Names

○ Adapter Reference

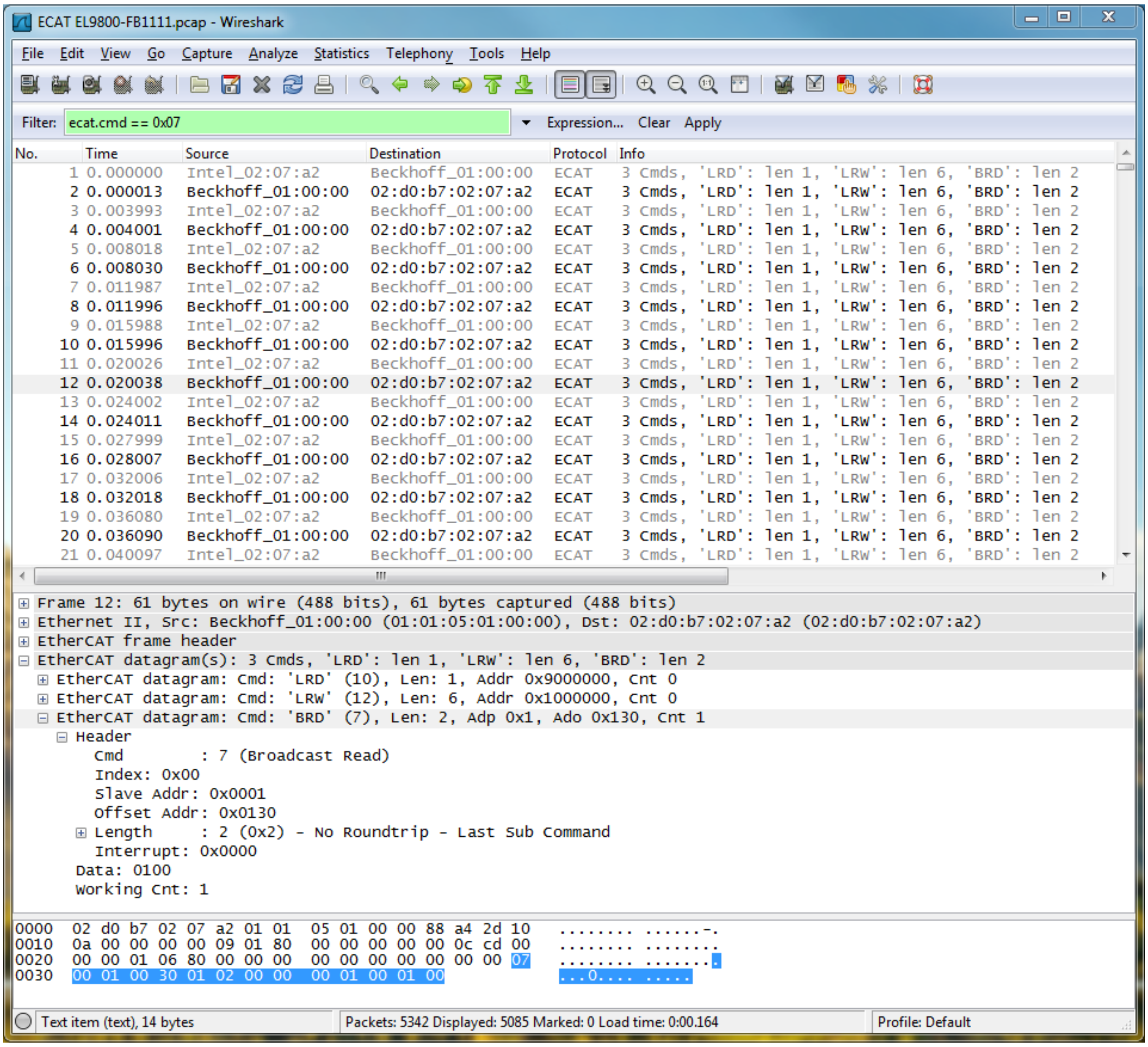
Adapter:

Freerun Cycle (ms): 4

Figure 22: TwinCAT Device Scan, Box Scan and Adapter Settings

2.4.3. Monitoring Communication and Network Diagnosis using Wireshark

In order to verify EtherCAT communication data, EtherCAT frames can be decrypted by a frame analyzing software such as Wireshark. Wireshark traces can be taken either on the EtherCAT master or via a real-time Ethernet probe. In order to record EtherCAT frames on a TwinCAT master the promiscuous mode needs to be activated (see Figure 22). The content of EtherCAT frames is displayed by Wireshark as shown below.


ECAT EL9800-FB1111.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: ecat.cmd == 0x07
Expression... Clear Apply

```
No.  Time      Source             Destination        Protocol Info
  1 0.000000  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  2 0.000013  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  3 0.003993  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  4 0.004001  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  5 0.008018  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  6 0.008030  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  7 0.011987  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  8 0.011996  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  9 0.015988  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 10 0.015996  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 11 0.020026  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 12 0.020038  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 13 0.024002  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 14 0.024011  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 15 0.027999  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 16 0.028007  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 17 0.032006  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 18 0.032018  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 19 0.036080  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 20 0.036090  Beckhoff_01:00:00  02:d0:b7:02:07:a2  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
 21 0.040097  Intel_02:07:a2     Beckhoff_01:00:00  ECAT   3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
```

```
⊞ Frame 12: 61 bytes on wire (488 bits), 61 bytes captured (488 bits)
⊞ Ethernet II, Src: Beckhoff_01:00:00 (01:01:05:01:00:00), Dst: 02:d0:b7:02:07:a2 (02:d0:b7:02:07:a2)
⊞ EtherCAT frame header
⊟ EtherCAT datagram(s): 3 Cmds, 'LRD': len 1, 'LRW': len 6, 'BRD': len 2
  ⊞ EtherCAT datagram: Cmd: 'LRD' (10), Len: 1, Addr 0x9000000, Cnt 0
  ⊞ EtherCAT datagram: Cmd: 'LRW' (12), Len: 6, Addr 0x1000000, Cnt 0
  ⊟ EtherCAT datagram: Cmd: 'BRD' (7), Len: 2, Adp 0x1, Ado 0x130, Cnt 1
    ⊟ Header
        Cmd        : 7 (Broadcast Read)
        Index: 0x00
        Slave Addr: 0x0001
        Offset Addr: 0x0130
      ⊞ Length     : 2 (0x2) - No Roundtrip - Last Sub Command
        Interrupt: 0x0000
      Data: 0100
      Working Cnt: 1
```

```
0000  02 d0 b7 02 07 a2 01 01  05 01 00 00 88 a4 2d 10   ........ ......-.
0010  0a 00 00 00 00 09 01 80  00 00 00 00 00 0c cd 00   ........ ........
0020  00 00 01 06 80 00 00 00  00 00 00 00 00 00 00 07   ........ ........
0030  00 01 00 30 01 02 00 00  00 01 00 01 00            ...0.... .....
```

Text item (text), 14 bytes | Packets: 5342 Displayed: 5085 Marked: 0 Load time: 0:00.164 | Profile: Default

Figure 23: Wireshark Screenshot

The EtherCAT dissector is already included in Wireshark.

Further detailed information about capturing, filtering and handling of Wireshark scans can be found within the EtherCAT Knowledge Base.

2.4.4. EtherCAT Conformance Test Tool

Besides of basic software and hardware debugging, in-house EtherCAT conformance testing is mandatory to verify that the device meets the EtherCAT communication requirements. Meeting this requirement is a minimum condition to sell the product as the EtherCAT compatible product. In-house EtherCAT conformance testing is done with the EtherCAT Conformance Test Tool (CTT).

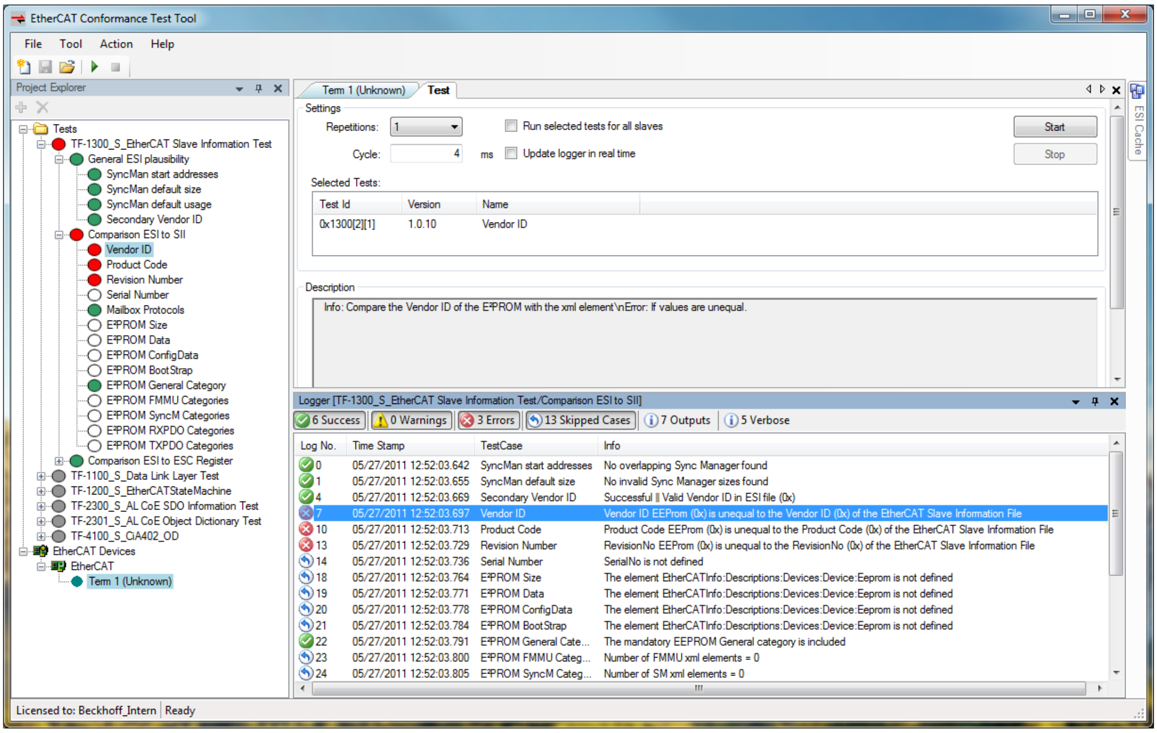
Application Note: To build a conformance test environment, the following items should be prepared.

- Windows PC and network card (100Mbit, full duplex and auto negotiation must be supported)
- In case the CU2508 is used a 1GBit/s network card is required
- CTT
- Download software via www.ethercat.org/cttdownload
- Get license subscription from Beckhoff (product name is ET9400) (see chapter 2.2.3).
- NOTE: Download and install the latest CTT version. The CTT is updated periodically
- Device under Test (DuT)
- EtherCAT Slave Information (ESI) file
- Packet analyzing software (e.g. Wireshark)
  A network probe might be useful when DuT support DCs
- Realt-time hardware extension

- The CTT runs on the Windows OS, which provides very limited real-time capabilities. To make real-time testing possible, e.g. for DuTs support DC, the hardware CU2508 (provided by Beckhoff) has to be used.

The ETG.7000.2 Conformance Test Record is a guideline for testing. Basically, proceed as follows.

- Install the CTT on the Windows PC
- Copy the ESI to the device descriptions folder in the local installation folder of the CTT
- Link the device to the Windows PC, start CTT and scan for the device to load it into the CTT
- Perform the tests provided by the CTT
- Update firmware, ESI, SII and everything else until all errors are gone. The CTT test logs help to understand where updates are necessary; see Figure 24 and the CTT documentation (Help file).



Settings

Repetitions: 1

☐ Run selected tests for all slaves

Cycle: 4 ms

☐ Update logger in real time

Selected Tests:

Test Id	Version	Name
0x1300[2][1]	1.0.10	Vendor ID

Description

Info: Compare the Vendor ID of the E^{2}PROM with the xml element\nError: If values are unequal.

Logger [TF-1300_S_EtherCAT Slave Information Test/Comparison ESI to SII]

6 Success | 0 Warnings | 3 Errors | 13 Skipped Cases | 7 Outputs | 5 Verbose

Log No.	Time Stamp	TestCase	Info
0	05/27/2011 12:52:03.642	SyncMan start addresses	No overlapping Sync Manager found
1	05/27/2011 12:52:03.655	SyncMan default size	No invalid Sync Manager sizes found
4	05/27/2011 12:52:03.669	Secondary Vendor ID	Successful ‖ Valid Vendor ID in ESI file (0x)
7	05/27/2011 12:52:03.697	Vendor ID	Vendor ID EEProm (0x) is unequal to the Vendor ID (0x) of the EtherCAT Slave Information File
10	05/27/2011 12:52:03.713	Product Code	Product Code EEProm (0x) is unequal to the Product Code (0x) of the EtherCAT Slave Information File
13	05/27/2011 12:52:03.729	Revision Number	RevisionNo EEProm (0x) is unequal to the RevisionNo (0x) of the EtherCAT Slave Information File
14	05/27/2011 12:52:03.736	Serial Number	SerialNo is not defined
18	05/27/2011 12:52:03.764	E^{2}PROM Size	The element EtherCATInfo:Descriptions:Devices:Device:Eeprom is not defined
19	05/27/2011 12:52:03.771	E^{2}PROM Data	The element EtherCATInfo:Descriptions:Devices:Device:Eeprom is not defined
20	05/27/2011 12:52:03.778	E^{2}PROM ConfigData	The element EtherCATInfo:Descriptions:Devices:Device:Eeprom is not defined
21	05/27/2011 12:52:03.784	E^{2}PROM BootStrap	The element EtherCATInfo:Descriptions:Devices:Device:Eeprom is not defined
22	05/27/2011 12:52:03.791	E^{2}PROM General Cate...	The mandatory EEPROM General category is included
23	05/27/2011 12:52:03.800	E^{2}PROM FMMU Categ...	Number of FMMU xml elements = 0
24	05/27/2011 12:52:03.805	E^{2}PROM SyncM Categ...	Number of SM xml elements = 0

Figure 24: Testing with the Conformance Test Tool

Conformance and interoperability are very important factors for the success of a communication technology. Conformance of the technology implementation with the specifications is the pre-requisite of interoperability, which means that devices of different manufacturers co-operate in the same networked application.

The conformance testing rules and policies according to the Vendor ID agreement are covered by the ETG.7000 Conformance Test Policy, available on the ETG website.

2.5. EtherCAT Product Labels and LEDs

It is recommended to consider the LEDs, device identification and captioning (e.g. of the ports) during the devices hardware design. This is subject of the ETG.9001 Marking Rules and the ETG.1300 Indicator Specification and Labelling Specification.

EtherCAT obligates various elements for indication. Such indication should be made as markings on the surface of EtherCAT slave box. The marking requirements are also the subject elements of the ETC conformance test (ETG.7000.2 Conformance Test Record).

Activity of EtherCAT devices is indicated by LEDs, which indicate the

- Current state of the state machine: Init, PreOp, SafeOp, Op (RUN LED)
- Error code (ERR LED)
- Link/Activity of the ports (L/A LED)

Application Note: Referring to the ETG.1300 Indicator and Labelling specification, the LEDs must work as shown in the following table.

Table 9: RUN and ERR LED Indications

RUN LED	EtherCAT State	ERR LED	EtherCAT State
Off	Init	Off	No Error
Blinking	Pre-Operational	Blinking	Invalid Configuration
Single Flash	Safe-Operational	Single Flash	Unsolicited State Change
		Double Flash	Application Watchdog Timeout
Flashes	Initialization or Bootstrap	Flickering	Booting Error
On	Operational	On	PDI Watchdog Timeout

Application Note: EtherCAT Branding. At least one of the following EtherCAT logos should show on the product or instruction manual:

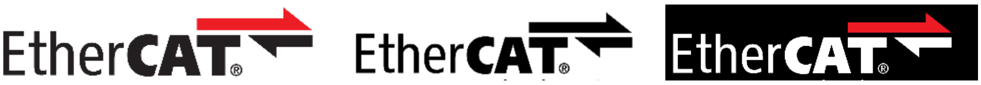


Figure 25: EtherCAT Product Branding Logos

The following English declaration of the EtherCAT trademark must appear in the instruction manual:

> *"EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany."*

Application Note: Requirements for port labels and L/A LED indication derived from the ETG.1300 Indicator and Labelling specification.

Table 10: Port and L/A LED Label Requirements

Label type	Requirements
IN port label	Must be placed near the port. The label should be clearly allocated to the subject port the characters on the label should be one of the following "In" or "ECAT IN" (Capitals and small characters both permitted).
OUT port label	Must be placed near the port. The label should be clearly allocated to the subject port. The characters on the label should be one of the following "Out" or "ECAT Out" (Capitals and small characters both permitted).
L/A LED label	Preferably the print characters should be placed directly next to the network interface but is not compulsory. The mark can be placed on other location or can be omitted. The print characters, if not omitted, should show one of the following phrases. "L/A", "Link/Act" or "Link/Activity" (Capitals and small characters both permitted). Label is required on removable connectors.

Application Note: Physical Connections.
A Schematic view of the sample circuit for PHYs is available in Beckhoff's PHY Selection Guide.

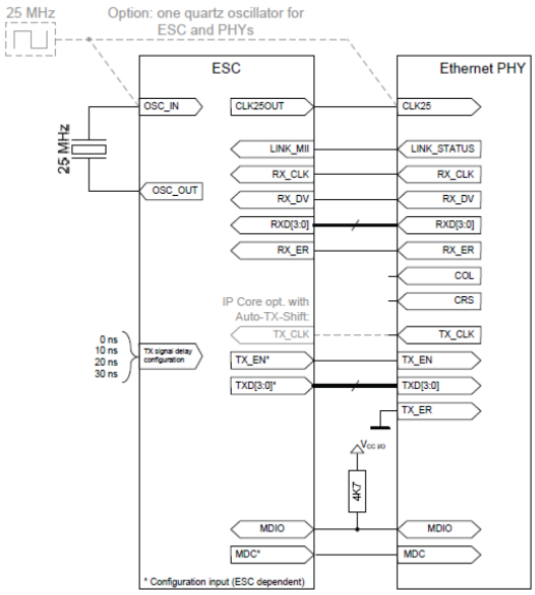


Figure 26: Connecting ESC and Ethernet PHY

When the EtherCAT slave is an enhanced or complex slave device using a µC, I/F of the µC and the ESC could be different with respect to the normal combination dependent on the type of ESC being used. Thus, careful observation is required to meet the expected performance.

As exemplary LVDS implementation an LVDS port connection is shown below. LDVS termination resistance should be placed near the signal input by each pair of received signals. In the figure, 100R denotes 100Ω.

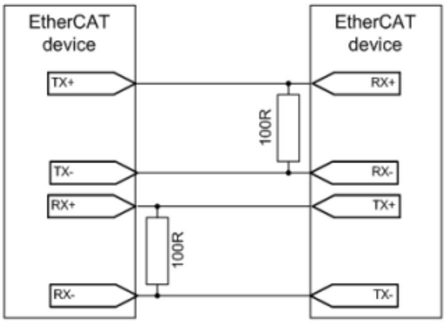


Figure 27: LVDS Connection

2.6. Official Conformance Test at an EtherCAT Test Center (ETC)

The procedure is described in detail in the ETG.7010 EtherCAT Conformance Guide. Following is an overview to the procedure of an official EtherCAT Conformance Test according to the ETG.9003 EtherCAT Conformance Test Policy.

1. Fill out the ETG.7000 Conformance Test Request form.
2. Send the request form to conformance@ethercat.org
    - When the test request is received, the ETC denoted in the test request starts arranging the test schedule and sends the test contract.
    - Return the signed contract by e-mail or FAX. The test fee invoice will not be issued unless the test contract is submitted.
    - Send out referenced test material. A device check list assists a reference for material which is to send to the ETC a week before the test.
    - Preparation of components to deliver. Ensure that all equipment is delivered to the ETC before the test date. It is NOT possible to deliver any missing items afterwards.
3. Test execution according to the ETG.7000.2 Conformance Test Record.
   If preferred to attend the test ensure to have a meeting arranged with the ETC.

4. Certification. By successfully passing the EtherCAT Conformance Test, a pass notice is issued by the ETG Headquarters. The “EtherCAT Conformance Tested” certificate will then be issued and sent to the device vendor.

2.7. Technical Support Tips

When having questions or problems with an EtherCAT device development, feel invited to engage individual support provided by the EtherCAT Technology Group (for contact, see chapter 3.1).

To optimize support processes, the following instructions lead to faster response time and improve support quality. Basically, explain the issue as detailed as necessary but as simple as possible.

- Which system architecture are you using
- Hardware components: ESC, µC, etc.
- Software components (& versions): Slave stack, master solution, etc.
- Infrastructure: topology, (self-made) cables, etc.
- Problem report:
- What: Can you shortly describe the behavior?
- When: Is the Problem reproducible?
- Where: Can you locate the problem?
- What was already tested?
- Additional information:
- Master configuration file, in suitable format
- E.g. *.tsm file (TwinCAT 2) or solution (TwinCAT 3)
- ESI files of involved device(s)
- Anything else that helps to process the issue
- screenshots
- log files
- In case of conformance testing, the Conformance Test Tool project file (*.ctp) with saved results
- Wireshark scan (*.pcap or *.pcapng format) capturing the problem. To focus the scan on necessary content, follow these instructions:
- Connect the smallest number of devices enabling to reproduce the problem
- Capture the network start-up phase, either in the same or a separate capture
- When using an Ethernet probe, report where exactly the probe was connected

3. EtherCAT Technology Group – Events and Support

The EtherCAT Technology Group (ETG) is the forum where key user companies from various industries and leading automation suppliers join forces to support, promote and advance the EtherCAT technology.

3.1. Basic Information about the ETG

Goals

EtherCAT is an open technology. The ETG stands for this approach and ensures that every interested company may implement and use EtherCAT.

At the same time the ETG aims to ensure the compatibility of EtherCAT implementations by defining functional requirements, conformance tests as well as certification procedures.

The ETGs goal is to ensure that EtherCAT technology meets and exceeds the requirements of the widest possible application range. In order to accomplish this goal the group combines leading control and application experts from machine builders, system integrators, end users and automation suppliers to provide both qualified feedback about application of the existing technology and proposals for future extensions of the specification.

The ETG organizes user and vendor meetings in which the latest EtherCAT developments are reviewed and discussed in regular periodical sessions.

Benefits for ETG Members

ETG members get preferred access to specifications, specification drafts, white papers, prototype evaluation products and initial batch products and thus have a head start in evaluating, using or implementing the EtherCAT technology.

The members are eligible to participate in technical working groups (TWG) and thus have influence on future enhancements of the EtherCAT technology specifications, like safety, conformance, and much more. A closer look to all available TWGs are found in the working group area on the ETG website.

The member companies may use the EtherCAT and the ETG logos to show their support for this technology.

How to join the ETG

If you are interested in becoming a member of the ETG, please contact the ETG headquarters for further information resp. membership request (see contacts following).

Membership Costs

The membership is free of charge, thus there are no annual membership fees. According the ETG by-laws a membership fee can only be introduced if the membership assembly decides so.

Technical Support

Technical support throughout the development process is provided by the ETG predominately by the headquarters in Germany, but also by the various ETG offices worldwide (depending on local capacity). If you need direct contact, please address your specific question to the ETG.

Before contacting ETG for support, we expect reading the mentioned documentation above as well as the recently listed information below (i.e. chapter 2.7). We strongly recommend visiting one of the EtherCAT workshops and/or seminars for developers when starting an EtherCAT implementation.

Also a good opportunity to ask for technical experience with EtherCAT and for technical questions is provided by the EtherCAT Forum and the EtherCAT Knowledge Base within the member section of the EtherCAT website.

Contact

ETG Headquarters
Email: info@ethercat.org
URL: www.ethercat.org

ETG Office North America
Email: info.na@ethercat.org

ETG Office China
Email: info@ethercat.org.cn

ETG Office Japan
Email: info.jp@ethercat.org

ETG Office Korea
Email: info.kr@ethercat.org

3.2. EtherCAT Training and Workshops from ETG Members

The following trainings are offered by the ETG. For the current agenda of EtherCAT workshops and trainings, consult the event section of the ETG website. ETG trainings are available for ETG members only and are free of charge.

Table 11: ETG Training

Part	Description
EtherCAT Training Class	In March & Sept., one day before the Technical Committee Meeting in Frankfurt (Germany).
EtherCAT Safety Training Class	Training focusing safety aspects (FSoE – Safety over EtherCAT)
EtherCAT Introduction for Salesmen	Training course for basics and key features of EtherCAT (on request)

Additionally, training is offered by vendors for product development, e.g. concerning slave implementation with the Beckhoff Evaluation Board (Table 12).

Table 12: EtherCAT Workshops from Vendors

Part	Description
EtherCAT Technology Basics for Developers (Beckhoff, TR8110)	One day training class handles: EtherCAT Basics Slave Structure Physical Layer Protocol Application Layer features including device profiles Distributed Clocks Device description in XML format (ESI) Master and slave implementation questions Overview standards and references
EtherCAT Evaluation Kit Workshop for Slave Developers (Beckhoff, TR8100)	One day hands-on workshop includes: EtherCAT hardware Installation of TwinCAT, incl. drivers Handling of PDI Slave Stack Code (SSC) ESC device overview (ET1100, IP Core) Device description in XML format (ESI)

Both workshops and training classes have proved to put the developer in a good starting position with a well-established understanding of the EtherCAT protocol, tools, development hardware and software including the Slave Sample Code as a basis to build the vendor specific application on top.

3.3. Plug Fests

Depending on the demand of ETG companies, Plug Fests are held several times a year. Every ETG member developing devices or tools with at least a functional prototype are allowed to attend. In practical tests interoperability and the latest features of the devices are tested and the EtherCAT Slave Conformance Test Tool is applied. Qualified feedback of EtherCAT specialists is provided.

Dates are published on the event section of the ETG website. An additional invitation email is automatically sent to the ETG representatives of the ETG member companies.

Participation at Plug Fests is free of charge. Attendees are not entitled to publish or communicate test results of other participating companies.

3.4. Official EtherCAT Conformance Test Certificate

An official EtherCAT Conformance Test is at option after successful in-house testing. With passing the EtherCAT Conformance Test successfully a “Conformance Tested” certificate is issued and thus, the vendor may label his device with the official conformance test mark and use the term for advertisement for the certified device exclusively.

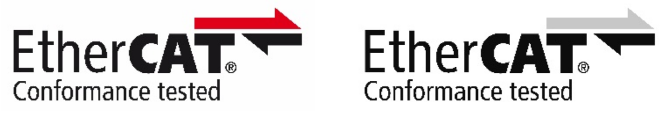


Figure 28: EtherCAT Conformance Test Logos

To apply for the EtherCAT Conformance Test at any EtherCAT Test Centre (ETC) send an Email to conformance@ethercat.org to ask for further information and the request form. On return of the request form to the ETG the requested ETC will contact you for further steps (see chapter 2.6).

The Conformance Guide explains the most important details on the topic and gives advice for preparation of the Conformance Test.

There are two official authentication test centers, one in Nuremberg, Germany, and one at ASTEM in Kyoto, Japan. The ETCs do not only perform the official conformance test, but also provide qualified feedback and implementation support for ETG members.

The official test performed by an ETC is referred as "EtherCAT Conformance Test" which is regarded as higher-level test above all other tests performed individually by the users (with the CTT) since interoperability and physical layer tests are covered as well.

When successfully passed the EtherCAT Conformance Test at an ETC, a notice is given to the ETG Headquarter. An “EtherCAT Conformance Tested” certificate is then issued and sent to the device vendor. Having obtained the Conformance Tested certificate the vendor may label the device with the official conformance test mark. The vendor may use the term for advertisement for the certified device exclusively.

The test fee differs by the ETCs. Please contact the ETC-Nuremberg directly for the test fee and further information. See the URL below for the list of ETCs and contact.

3.5. Technical Committee

The TC serves as central technical board. It establishes working groups, task forces and receives their reports. Other duties of the TC are to inform about enhancements of the EtherCAT technology, progress on standardization and to discuss current technical issues with the attending ETG members.

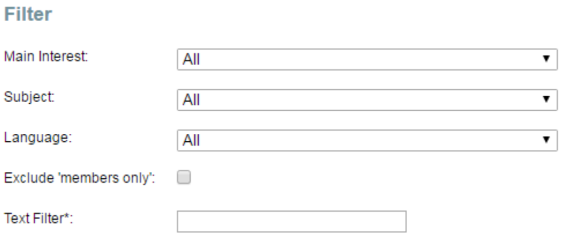
Dates are published on the event section of the ETG website. An additional invitation email is automatically sent to the ETG representatives of the ETG member companies.

Participation at the TCs is free of charge.

3.6. Information and Support

3.6.1. Download Area on the web site

There is heaps of information available within the download area of the EtherCAT web site at www.ethercat.org/downloads. Take advantage of the filter options, too:



Filter

Main Interest:	All ▾
Subject:	All ▾
Language:	All ▾
Exclude 'members only':	□
Text Filter*:	

Furthermore, the filter can be set by using URL-Parameter “?tf=” directly, for example:

http://www.ethercat.org/en/downloads.html?tf=diagnosis

http://www.ethercat.org/en/downloads.html?tf=safety

http://www.ethercat.org/en/downloads.html?tf=conformance

3.6.2. Knowledge Base

As one of the main sources to complement the EtherCAT specifications the Knowledge Base (www.ethercat.org/kb) provides:

- Glossary:
  description of EtherCAT terms including references to other related readings
- “How To …?” descriptions:
  description of how to e.g. make a networks scan, test CoE communication and many other things
- Descriptions:
  elaborating the specifications where necessary

The Knowledge Base is continuously updated based on the questions we receive at the ETG team – so make it a habit to check on the Knowledge Base first. Also, if you are missing any information, please help us with your input on possible new entries on the Knowledge Base and send it to info@ethercat.org.

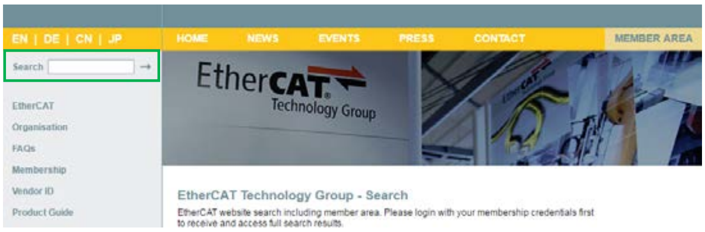
3.6.3. Developer’s Forum

On the EtherCAT Forum, every ETG member is invited to discuss the EtherCAT technology and to post own questions. Many practical questions are already answered in the following forum topics:

- EtherCAT Specifications
- Proposals
- Implementing EtherCAT
- Master and Slave Devices
- Evaluation Kit Hardware and Software
- EtherCAT Slave Conformance Test
- Test Cases
- Slave Conformance Test Tool
- EtherCAT Technology Group
- ETG Services
- New Downloads
- EtherCAT.org Website
- Suggestions for improvements and comments

3.6.4. Search the EtherCAT web site

A search field is always accessible when surfing the EtherCAT web site in the upper left corner or via www.ethercat.org/search.



EtherCAT Technology Group - Search

EtherCAT website search including member area. Please login with your membership credentials first to receive and access full search results.

3.6.5. Technical Support

When having questions during your EtherCAT device development and you just cannot find the right answer on any of the above mentioned sources, engage with ETG’s technology experts directly. For support tips, see chapter 2.7.

EtherCAT® and EtherCAT® P
Slave Implementation Guide

SECTION II – EtherCAT Development Components

EtherCAT Development Products, Evaluation Kits, Slave Controllers, Communication Modules, Implementation Specifics

EtherCAT® Technology Group

CONTENTS

TABLES

FIGURES

1. Introduction

There is a wide range of EtherCAT Slave implementation possibilities available, accounting for the different types of devices, as well as for the different types of development approaches and needs.

One outstanding feature of EtherCAT is the number of ESC vendors, ranging from different ASCIS, to multi-protocol-solutions, SoCs combining an ESC and microcontroller or even CPU on a single silicon and communication modules with an API. Some already come with an EtherCAT slave stack, other support the integration of widely used state-of-the arts stacks.

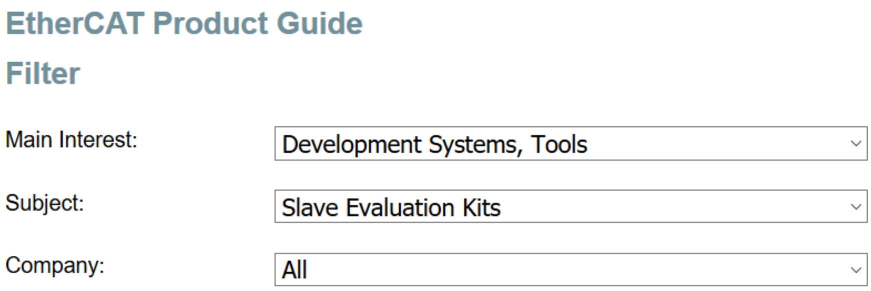
Vendors provide development kits including documentation and for some specific trainings are offered.

This section lists those devices and products. Lists might not complete, however, they already provide an extensive overview. ETG encourages vendors to add their product to it.

2. EtherCAT Slave Evaluation Boards

Evaluation boards offer a good starting point for EtherCAT slave development. They provide the hardware for a “sample EtherCAT slave”, so that software development can be started. The product-specific PCB design can be initiated in parallel.

This chapter provides an overview of EtherCAT slave development boards while it cannot claim to be complete. For latest information visit the EtherCAT product guide with the filter settings as shown in Figure 29 and in a next step, the vendors directly.



EtherCAT Product Guide

Filter

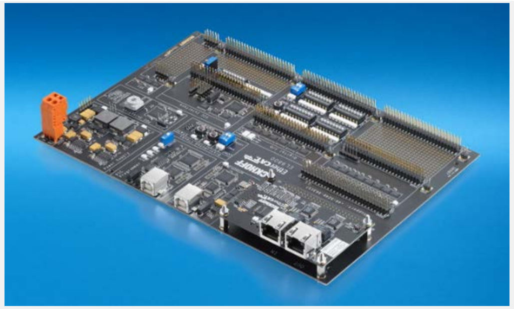
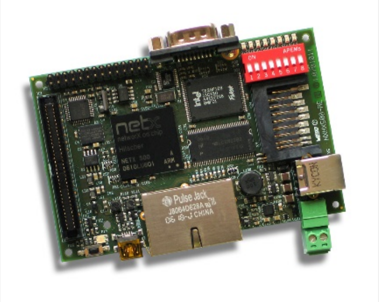
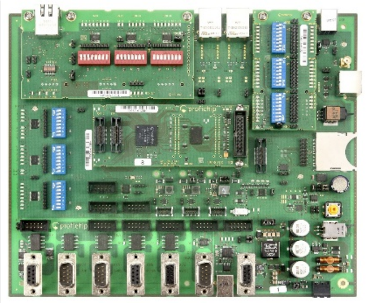
Main Interest: Development Systems, Tools

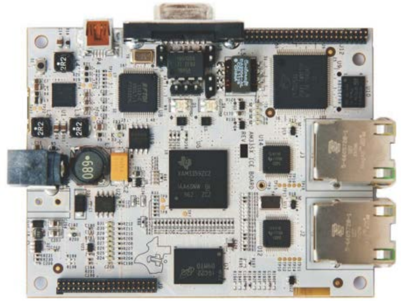
Subject: Slave Evaluation Kits

Company: All

Figure 29: Product Guide filter for Evaluation Boards

Table 13: EtherCAT Evaluation Kits

Beckhoff EL9820	**Hilscher NXHX 500**	**Profichip ANTAIOS**
		
For the Evaluation Kit (base board EL9800 with EtherCAT piggyback controller board), a one-day hands-on workshop and a preceding one day training class explaining the EtherCAT protocol are offered (section I, chapter 3.2). Base board with • Socket for FB1111 EtherCAT Piggyback Board with EtherCAT Slave Controller • Several PDI (32 Bit Digital I/O, 8/16-bit µC, SPI) to connect hardware • On-board PIC connected via SPI to ESC with pre-installed SSC • Debugger Interface for MPLAB® • Power supply (24V) • Cables, Documentation • EtherCAT Slave Stack Code (SSC) • Handling of synchronous and asynchronous data exchange via DPRAM • Support of mailbox protocols (CoE incl. Object Dictionary, EoE, FoE, AoE) • Support of synchronized application using Distributed Clocks • Slave Controller Board, equipped with ET1100	The netX network controller with its 32 Bit / 200 MHz ARM CPU provides a high degree of computing performance and comprehensive peripheral functions for single chip solutions in price-sensitive applications. Here the network protocols and the application program together use the resources of the netX and are carried out together in a Real-Time operating system. • I/O, parallel host interface, UART, USB, JTAG • DIP-switches and LEDs for I/O, SD card slot, fieldbus interface (optional), Multi-protocol support • EtherCAT Slave Hardware Abstraction Layer (HAL) available on demand • integrated debug interface supplied with the HiTOP development environment from Hitex.	The new ANTAIOS multi-fieldbus controller from profichip enables higher integration in a smaller package to fulfill the requirements of the industrial automation market. The latest development by profichip combines a powerful processor for advanced user applications with an effective and very flexible communication technology for today's industrial applications demanding sophisticated real-time capabilities. Additionally a unique feature of this System-On-Chip (SoC) is the high-speed backplane communication master for profichip's SliceBus® Technology. • ARM® Cortex®-A5 (288MHz, 32kB+32kB cache, 64Bit FPU) • 2 Port Real-Time Ethernet Switching Unit (Profinet IRT, EtherCAT, etc.) • 2 integrated Ethernet Phys (copper + fiber) • 1 GBit Ethernet MAC • Also non ethernet based fieldbus protocols are supported • DDR2 external memory interface • QuadSPI interface (e.g. NOR Flash for firmware) • SD/MMC, NAND, USB2 device, SRAM master/slave, SPI master/slave
https://www.beckhoff.de/english.asp?ethercat/el9820_el9821_el9830_el9840_el9803.htm	https://www.hilscher.com/products/product-groups/network-controller/development-boards/?	http://www.profichip.com/products/tools/evaluation-kits/antaios-evaluation-kit/

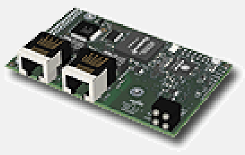
TI AM3359 Industrial Communications Engine	**TESSERA Technology R-IN32M3-EC**	
		
The ICE provides development of industrial communication applications, i.e. communication modules, I/O devices, sensors and other similar applications. It includes the essential peripherals for the EtherCAT communication and further industrial communication standards. The SDK includes a SYS/BIOS™ based real-time kernel with application-level communication stack and device drivers. The development and debug tool chain is also included with this platform. • Sitara AM3359 ARM Cortex-A8 MPU • RJ-45 connected to TLK110 Ethernet PHY • 8 Digital In, 8x Digital Out • 8 MByte Serial SPI Flash • 256 MByte DDR2 (opt.) • 8 kByte Dual-port RAM • Micro-SD slot • CAN, SPI, GPIO and UART • Parallel I/O to dual port RAM • JTAG via USB port (optionally 20 pin JTAG header) • Debug UART via USB port	TS-R-IN32M3-EC is the platform to develop software and system for Renesas Electronics industrial Ethernet communication LSI, R-IN32M3-EC. EtherCAT and other industrial protocol can be evaluated. Sample software can be downloaded in the Renesas Electronics Web site or protocol stack vendor Web site. • Renesas Electronics “R-IN32M3-EC” • ARM Cortex™-M3 with HW-RTOS • 1.3MByte embedded SRAM (100MHz) • 10Base-T/100Base-TX Ethernet PHY(2ch) • Flash memory (Serial) : 32Mbit×DualFlash memory (Parallel) : 2M×16bit • EEPROM : 16Kbit • UART(USB), CSI, I2C, CAN, RJ45 (Ethernet/EtherCAT) • Extension connector, GPIO (DIP, LED, pin header), CC-Link • 20pin half pitch connector (Trace supported) • IAR Systems : I-jet/JTAGjet	
http://www.ti.com/tool/TMDSICE3359?keyMatch=TMDSICE3359&tisearch=Search-EN-Everything	www.tessera.co.jp/eng/products/r-in32m3-ec-e.html	

3. EtherCAT Slave Communication Modules

An overview to a selection of communication modules is given first. Several of the communication modules are then listed in alphabetical order with detailed description.

3.1. EtherCAT Communication Modules Overview

Table 14: EtherCAT Communication Modules

	COMX	**ANYBUS-S**	**ANYBUS-CC**	**FB1111-0140 | 0141 | 0142**	**UMD2**
					
Hardware Supplier	Hilscher	HMS		Beckhoff	OKI
Size (mm)	70 x 30 x 18	54 x 86 x 16,6	51,8 x 50,1 x 22,3	55 x 85,5 x 14	20 x 30 x 4
ESC*	NetX 100	FPGA with IP-Core + ASIC		ET1100	SH2A+ET1100
µC Interface	DPM (8/16bit)	DPM (8bit)		8/16bit BUS SPI 32bit Digital I/O	
LEDs (RUN/ERR/LINK)	All	All		All	RUN, ERR
No. of Ports	2 x RJ45	2 x RJ45		2 x RJ45	2 x (MII/EBUS)
Power Supply	3.3V	5V	3.3V	5V	5V
Power Consumption	700mA	450mA	500mA	700mA	200mA
Further Information	chapter 3.3	https://www.anybus.com/products/embedded-index/embedded-networks/ethercat		chapter 3.2	

*see chapter **Error! Reference source not found.** for detailed information about available EtherCAT features which are depending on the applied ESC.

3.2. Beckhoff FB1111

The FB1111 EtherCAT piggyback controller boards offer complete EtherCAT connection based on the ET1100 EtherCAT ASIC. All FB1111 have the same form factor and can be used with the EL9800 EtherCAT Evaluation Kit. They can be integrated as EtherCAT interfaces in devices.

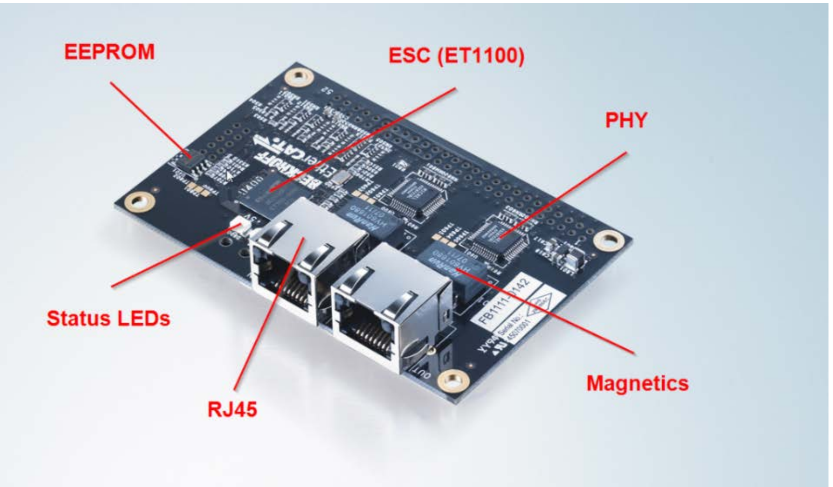


Figure 30: Beckhoff FB1111

Table 15: FB1111 Options

Part	Description
FB1111-0140	EtherCAT piggyback controller board with ET1100 (ASIC) and µC interface; can be integrated as EtherCAT interface in devices.
FB1111-0141	EtherCAT piggyback controller board with ET1100 (ASIC) and SPI interface; can be integrated as EtherCAT interface in devices.
FB1111-0142	EtherCAT piggyback controller board with ET1100 (ASIC) and digital I/O interface; can be integrated as EtherCAT interface in devices; included in the EL982x evaluation kit and together with the delivered adapter card EL9803 all interfaces (µC, SPI, digital I/O) can be applied. This is the most flexible solution for starting an EtherCAT implementation.

Further information: www.beckhoff.com/english.asp?ethercat/fb1111_fb1122_fb1130.htm

3.3. Hilscher comX

- Interfaces: Host processor over dual-ported memory (parallel)
- Ports: 2 (100BASE-TX)

All stacks are implemented as slave protocols and are executed on the comX-Module. Data exchange with the host application is carried out via Dual-Port-Memory interface. The process data images are available directly via memory read and write functions. The comX Module features two RJ45 connectors for Ethernet. netX based comX-Modules gets its identity by loading an appropriate firmware file.

- All Real-Time-Ethernet System use netX Network Controller
- Available as Master and Slave
- Two Ethernet Ports with Switch and Hub for Line Topology
- System/Status/Link/Activity LEDs
- 8 or 16-Bit Host Application Interface
- USB & UART Diagnostic Interface
- Direct Process Data Access
- Same Dimensions and Pin Compatible like our well-known COM-C Module
- SYCON.net as configurator based on FDT/DTM
- Short 'Time-To-Market'

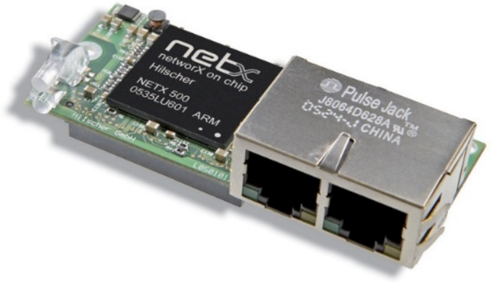


Figure 31: Hilscher comX module

Further information: https://www.hilscher.com/products/product-groups/embedded-modules/communication-module/

4. EtherCAT Slave Controller (ESC) Variants

The following table gives an overview about several ESC variants. Concerning to a continuing development of products by ETG-Members we recommend also to have a look on the online version of the EtherCAT Slave Controller (ESC) Overview.

Table 16: ESC Overview

Name/Link	ET1100	ET1810/ET1811/ ET1812	ET1851/ET1816/ ET1817	netX 100	netX 500	netX 50	netx51	netX52
Type	ASIC	Altera FPGA + IP Core	Xilinx FPGA + IP Core	ASIC	ASIC	ASIC	ASIC	ASIC
Supplier	BECKHOFF	BECKHOFF	BECKHOFF	hilscher COMPETENCE IN COMMUNICATION	hilscher COMPETENCE IN COMMUNICATION	hilscher COMPETENCE IN COMMUNICATION	hilscher COMPETENCE IN COMMUNICATION	hilscher COMPETENCE IN COMMUNICATION
Package	BGA128 0.8 mm pitch	FPGA dependent	FPGA dependent	BGA345 1 mm pitch	BGA345 1 mm pitch	PBGA324 1 mm pitch	PBGA324 1 mm pitch	PBGA244 1 mm pitch
Size	10 x 10 mm	FPGA dependent	FPGA dependent	22 x 22 mm	22 x 22 mm	19 x 19 mm	19 x 19 mm	15 x 15 mm
µC Interface	serial/parallel (8/16bit, sync/async)*	serial/parallel (8- /16-bit, async) AVALON®*	serial/parallel (8- /16-bit, async) OPB®* and PLB®*	µC bus (internal, 32bit)	µC bus (internal, 32bit)	µC bus (internal, 32bit)	µC bus (internal, 32bit)	µC bus (internal, 32bit)
Digital I/O	8-32*	8-32*	8-32*	-	-	-	-	-
General Purpose I/O	0-32*	0-128*	0-128*	16	16	32	32	24
DPRAM	8 kByte	0...60 kByte*	0...60 kByte*	256/512 Byte (Mailbox/Process Data)	256/512 Byte (Mailbox/Process Data)	6 kByte	6 kByte	6 kByte
SyncManager Entities	8	0…8*	0…8*	4	4	8	8	8
FMMU Entities	8	0…8*	0…8*	3	3	8	8	8
Distributed Clock Support	yes	yes*	yes*	yes	yes	yes	yes	yes
No. of Ports	2-4 (MII/E-BUS)*	1-3 (MII/max. 2 RMII)	1-3 (MII/max. 2 RMII)	2 (100BaseTX)	2 (100BaseTX)	2 (100BaseTX)	2 (100BaseTX)	2 (100BaseTX)
Specials	BGA routable with standard PCB	Various license models and OpenCore Plus are available. A wide range of Altera FPGAs are supported	Various license models and evaluation Version are available. A wide range of Xilinx FPGAs are supported	Multi-protocol support, Integrated PHYs, Integrated µC (ARM9-200MHz)	Multi-protocol support, Integrated PHYs, Integrated µC (ARM9-200MHz)	Multi-protocol support, Integrated PHYs, Integrated µC (ARM9-200MHz)	Multi-protocol support, Integrated PHYs, Integrated µC (ARM9-100MHz)	Multi-protocol support, Integrated PHYs, Integrated µC (ARM9-100MHz)

Table 16: ESC Overview (forwarding)

Name/Link	netX4000	netX90	Anybus NP40	XMC4300	XMC4800	fido5200	LAN9252	ANTAIOS	EC-1
Type	ASIC	ASIC	ARM MPU	ARM MPU	ARM MPU	ASIC	ASIC	ARM MPU	ARM MPU
Supplier	hilscher COMPETENCE IN COMMUNICATION	hilscher COMPETENCE IN COMMUNICATION	HMS	infineon	infineon	Innovasic	MICROCHIP	profichip automation in silicon	RENESAS
Package	PBGA596 1 mm pitch PBGA420 1mm pitch	LFBGA144 0.8 mm pitch	BGA VF400 0.8 mm pitch	100 LQFP (0.5 mm)	100 LQFP (0.5 mm)	144 FBGA (0.8 mm) 144 LQFP (0.5 mm)	64 pin QFN (0.5 mm pitch) 64 pin TQFP-EP (0.5 mm pitch)	TFBGA-380 (0.65 mm pitch) TFBGA-385 (0.8 mm pitch)	196 pin BGA (0.8 mm)
Size	27 x 27 mm 23 x 23 mm	10 x 10 mm	17 x 17 mm	16 x 16 mm	20 x 20 mm 16 x 16 mm 12 x 12 mm	10 x 10 mm (FBGA) 20 x 20 mm (LQFP)	9 x 9 mm 12 x 12 mm	15 mm x 15 mm 19 mm x 19 mm	12 x 12 mm
µC Interface	µC bus (internal, 64bit)	µC bus (internal, 32bit)	Anybus interface (8- / 16-bit 30 ns parallel, 20 MHz SPI, Shift register, UART)	µC bus (internal, AHB)	µC bus (internal, AHB)	parallel (16/32-bit)	Host Bus/SPI/SQI	SPI / QSPI / 16 Bit asynchronous interface	USB Host/Function, CAN, SCIFA, I2C RSPI, Flash
Digital I/O	-	-	256 / 256 (Shift register mode)	-	-	8 (Event Output/Capture)	0-16*	-	-
General Purpose I/O	106	16	-	0 - 46	0 - 123	-	0-16*	up to 32	115* GPIOs / 8 Input (port multiplexed, partial 5V-tolerant, open drain, input pull-up)
DPRAM	6 kByte	6 kByte	12 kByte	8 kByte	8 kByte	10 kByte	4 kByte	up to 32 kByte	512 KB (ATCM) with ECC 32 KB (BTCM) with ECC
SyncManager Entities	8	8	4	8	8	8	4	8	8
FMMU Entities	8	8	4	8	8	8	3	8	8
Distributed Clock Support	yes	yes	yes	yes (64 Bit)	yes (64 Bit)	yes	yes	yes (64 bit)	yes (64 bit)
No. of Ports	2 (100BaseTX)	2 (100BaseTX)	2 (MII)	2 (MII)	2 (MII)	2 (MII/RMII)	2 (100BaseTX) + opt. 1 (MII)	2 (100BaseTX) or 2 (MII)	2 (MII)

Name/Link	netX4000	netX90	Anybus NP40	XMC4300	XMC4800	fido5200	LAN9252	ANTAIOS	EC-1
Specials	Multi-protocol support, Integrated PHYs, Integrated µC (ARM Cortex R7-400MHz) Additional integrated Application Controller (ARM Cortex A9 Dual Core - 600 MHz)	Multi-protocol support, Integrated PHYs, Integrated µC, OnChip Flash 1,5 Mbytes, OnChip DC-DC Converter, (ARM Cortex M4-100MHz) Additional integrated Application Controller (ARM Cortex M4 - 100 MHz)	Multi-protocol support, ESC Frame forwarding delay: 114 ns, MDP, possible to implement several device profiles	EtherCAT® node on an ARM® Cortex®-M4 processor with up to 256kB on-chip flash, 128kB on-chip RAM and analog/mixed signal capabilities. Qualified for up to 125°C ambient temperature.	EtherCAT® node on an ARM® Cortex®-M4 processor with up to 2MB on-chip flash, 352kB on-chip RAM and analog/mixed signal capabilities. Qualified for up to 125°C ambient temperature.	Multi-protocol support, Gigabit Ethernet support, Event Output/Capture I/O	Cable Diagnostics, 100FX support, 2 integrated PHYs, integrated 1.2V regulator	Multi fieldbus protocol support, 2 x integrated PHYs, 1 x integrated GBit Ethernet MAC, Integrated ARM® Cortex®-A5 (288MHz), Backplane communication: SliceBus master for profichip's SNAP+ ASIC, Integrated technology module (2xSII / 4xPWM / 4xCounter), QuadSPI interface (e.g. NOR Flash for firmware), DDR2 external memory interface, Other external interfaces: SD/MMC, NAND, USB2 device, SRAM master/slave, SPI master/slave	Safety Functions, Multi-Function Pin Controller

Table 16: ESC Overview (forwarding)

Name/Link	RZ/T1	R-IN32M3-EC	Sitara AM3357/9	Sitara AM4377/9	Sitara AM571xE	Sitara AM572xE	Sitara AMIC110 SoC	TMC8460	TMC8461
Type	ARM MPU	ARM MPU	ARM MPU	ARM MPU	ARM MPU	ARM MPU	ARM MPU	ASIC	ASIC
Supplier	RENESAS	RENESAS	TEXAS INSTRUMENTS	TEXAS INSTRUMENTS	TEXAS INSTRUMENTS	TEXAS INSTRUMENTS	TEXAS INSTRUMENTS	TRINAMIC MOTION CONTROL	TRINAMIC MOTION CONTROL
Package	FBGA320 0.8 mm pitch	BGA324 1 mm pitch	324NFBGA 0.8 mm pitch	491 BGA, 0.65mm pitch (0.8 mm effective routing)	760 BGA 0.8 mm pitch	760 BGA 0.8 mm pitch	324NFBGA 0.8mm pitch	VFGG400 Very Fine Pitch BGA 0.8 mm pitch	BGA144 0.8 mm pitch
Size	17 x 17 mm	19 x 19 mm	15 x 15 mm	17 x 17 mm	23 x 23 mm	23 x 23 mm	15x15mm	17 x 17 mm	10x10 mm

Name/Link	RZ/T1	R-IN32M3-EC	Sitara AM3357/9	Sitara AM4377/9	Sitara AM571xE	Sitara AM572xE	Sitara AMIC110 SoC	TMC8460	TMC8461
µC Interface	16/32-bit parallel and various serial (SPI/I2C/UART)	16/32-bit parallel (master/slave) and serial (SPI/I2C/UART)	200 MHz interconnect (internal, 32bit)	200 MHz interconnect (internal, 32bit)	200 MHz interconnect (internal, 32bit)	200 MHz interconnect (internal, 32bit)	200 MHz interconnect (internal, 32bit)	SPI or PDI-emulation	SPI or PDI-emulation
Digital I/O	-	-	8	8	8	8	8	0 (see specials)	0..16*
General Purpose I/O	0-209*	0-96*	>32	>32	>32	>32	>32	0 (see specials)	0..24*
DPRAM	8 kByte	8 kByte	32 kByte	32 kByte	32 kByte	32 kByte	32 kByte	16 kByte	16 kByte
SyncManager Entities	8	8	8	8	8	8	8	6	8
FMMU Entities	8	8	8	8	8	8	8	6	
Distributed Clock Support	yes	yes	yes	yes	yes	yes	yes	yes (64 Bit)	
No. of Ports	2 (RMII/MII)	2 (100BaseTX)	2 (MII)	2 (MII)	4 (MII)	4 (MII)	2 (MII)	2 (MII)	2 (MII)
Specials	Additional Ethernet port (RMII/MII), 2-axis high-speed motion control support, digital encoder interfaces (EnDat, BiSS, others), Multi-protocol support, security option, functional safety support, Cortex-R4F (450/600MHz), Cortex-M3 (150MHz) cores	Multi-protocol support, SPI, I2C, UART, 1.3 Mbyte int. RAM, <1W typical incl. 2 PHYs	Industrial Communications Subsystem (PRU-ICSS) for multi-protocol support, Gigabit Switch, CAN, display, ARM Cortex-A8 (275MHz-1000MHz)	Multi-protocol support, Second PRU-ICSS for Motor control (EnDat, sigma delta filtering etc), Gigabit Switch, CAN, Display subsystem, 2D/3D graphics, Camera I/F, Optional secure boot, ARM Cortex-A9 (upto 1 GHz)	Dual Industrial Communications Subsystem (PRU-ICSS) for multi-protocol support (2 EtherCAT slave instances or EtherCAT slave to protocol gateway), Motor control (EnDat, sigma delta filtering), 2D/3D Graphics, Display subsystem, Video acceleration, PCIe, SATA, Optional secure boot, ARM Cortex-A15 (upto 1.5GHz), 2x M4 cores, 1x C66x DSP core	Dual Industrial Communications Subsystem (PRU-ICSS) for multi-protocol support (2 EtherCAT slave instances or EtherCAT slave to protocol gateway), 2D/3D Graphics, Display subsystem, Video acceleration, PCIe, SATA, Optional secure boot, 2x ARM Cortex-A15 (upto 1.5 GHz), 2x M4 cores, 2x C66x DSP cores	Entire EtherCAT slave controller can be implemented on internal memory (no external DDR needed), Industrial Communications Subsystem (PRU-ICSS) for multi-protocol support, CAN	MFCIO block - additional multi-function and control block in hardware for complex real-time IO (GPIO, Watchdog, SPI Master, ABN interface, S/D output, 3ph PWM), direct MCU control or accessible via DPRAM/SyncManagers	Wide supply range (up to 35V), 2x Integrated DC/DC regulators, 8x Direct High Voltage I/Os, Multi-function I/O block, BGA routable with standard PCB

5. Missing your device?

Section II generally contains a snapshot of the spectrum of available products for a slave implementation. ETG members that offer EtherCAT development products, EtherCAT implementation services and EtherCAT workshops are invited to contribute information to the ETG for this guide. A range of products can be found at www.ethercat.org/products.

If you are missing your device here, you found an error or you have a suggestion for slave implementation support, feel free to contact the ETG and help to improve this document.

EtherCAT and EtherCAT P
Slave Implementation Guide

SECTION III – EtherCAT P Introduction and Implementation

EtherCAT P Technology Introduction, EtherCAT P Specifications and Documents, Licensing, Conformance Testing, Implementation

CONTENTS

TABLES

FIGURES

1. EtherCAT P Introduction

The following chapter describes the EtherCAT P technology and its benefit in brief. It provides an overview, however, it does not mean to replace reading the EtherCAT and EtherCAT P specifications and documents.

1.1. What is EtherCAT P

EtherCAT P is an enhancement to EtherCAT: it combines power (2 x 24V/3A) and the EtherCAT data transmission on the same four wires (Figure 34).

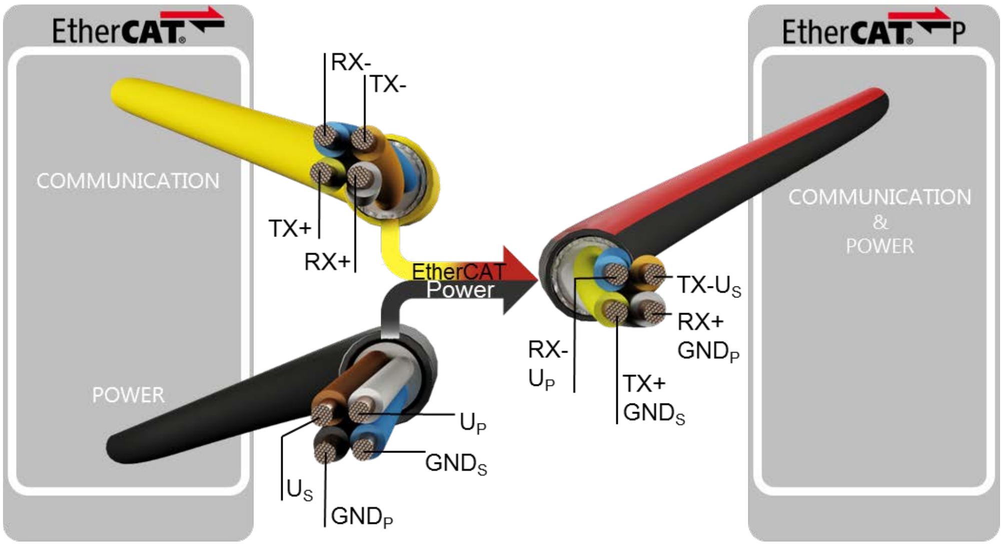


Figure 32: EtherCAT P – power combined with EtherCAT on the same four wires

As shown in Table 17 the four wires utilized for EtherCAT and the four wires used for powering U_S (for logic) and U_P (for the output peripherals) are combined on the same four wires by EtherCAT P.

Table 17: Combining DC power and communication high frequency signals on the same four wires

(typical) Wire Color	Yellow	Orange	White	Blue
EtherCAT	TX+	TX-	RX+	RX-
Power	GND_S	U_S	GND_P	U_P

The Ethernet signal used for EtherCAT is combined with the DC currents for U_S and U_P and provides a technology that comprises the following main features:

- Dual power supply
- U_S for system and sensors, 24 V DC/3 A
- U_P for peripheral voltage for actuators, 24 V DC/3 A
- Power forwarding through EtherCAT P devices within each network topology (e.g. Daisy-chain, Line, etc.)
- 100 % EtherCAT-compatible
- 100 Mbit/s full duplex, “processing on the fly”, Distributed Clocks, etc.
- Cascadable in all topologies (star, line, tree)

1.2. EtherCAT P Connectors and Cables

EtherCAT P connectors for 24V/3A are M8 P-coded connector. This connector provides a unique mechanical keying. This prevents from accidentally connecting EtherCAT devices to an EtherCAT P device. As a result of this simple mechanical concept no smart chips for power-sensing inside the

EtherCAT devices are required. Figure 33 shows the M8 P-coded connector. EtherCAT P cable colors are specified to be black and red.

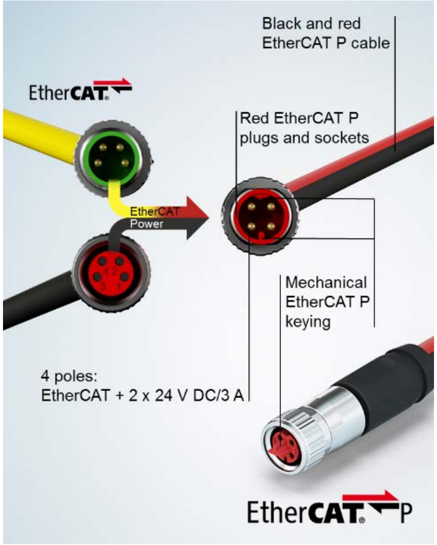


Figure 33: M8 P-coded connector and cable

In combination with the M8 P-coded connectors certain cables are specified (e.g.AWG22/7 and AWG24/7).

When an EtherCAT P cable is combined with an additional power-cable in a hybrid cable, a trapezoidal EtherCAT P connector is used. This allows for a high-density packaging within the hybrid cable as shown in Figure 34.

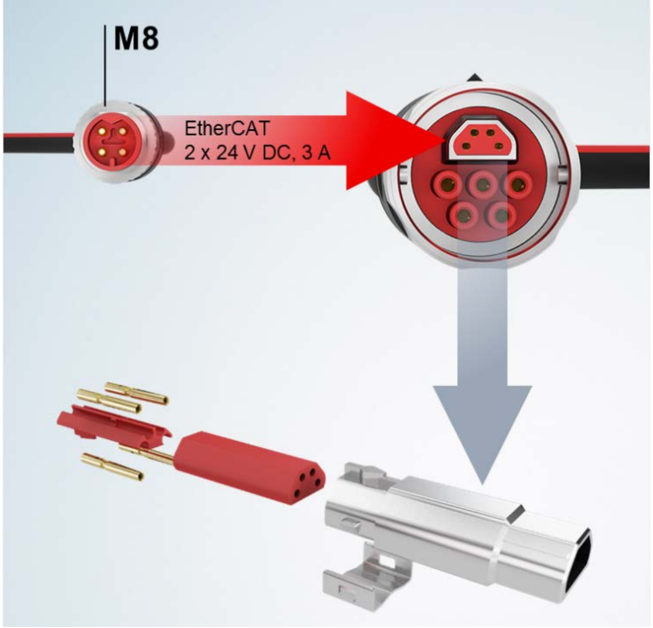


Figure 34: Trapezoidal EtherCAT P connector for hybrid cables

The hybrid cables allows additional energy transmission to supply complete machines, cabinets, robots with one cable including power and EtherCAT for example.

EtherCAT P hybrid cabling and connector technology is under development and will be added to the ETG specifications at a later stage.

1.3. PHY Selection

A list of recommended PHYs is provided by the Application Note – PHY Selection Guide,
Due to the internal interconnection, EtherCAT P places an increased requirement on the slave's analog circuitry design, including its PHYs.
An initial assessment of the slave, specifically the PHY's behavior already used in an existing slave implementation can be done as described in chapter 3 of this section.

1.4. EtherCAT P Use-Cases

EtherCAT P combines all the EtherCAT features – such as line/tree/star topology, unlimited number of devices in the network, Distributed Clocks, diagnosis features, fast EtherCAT performance, and more - with power on the same cable and connector.

It suites for all different kind of devices as shown in Figure 5.

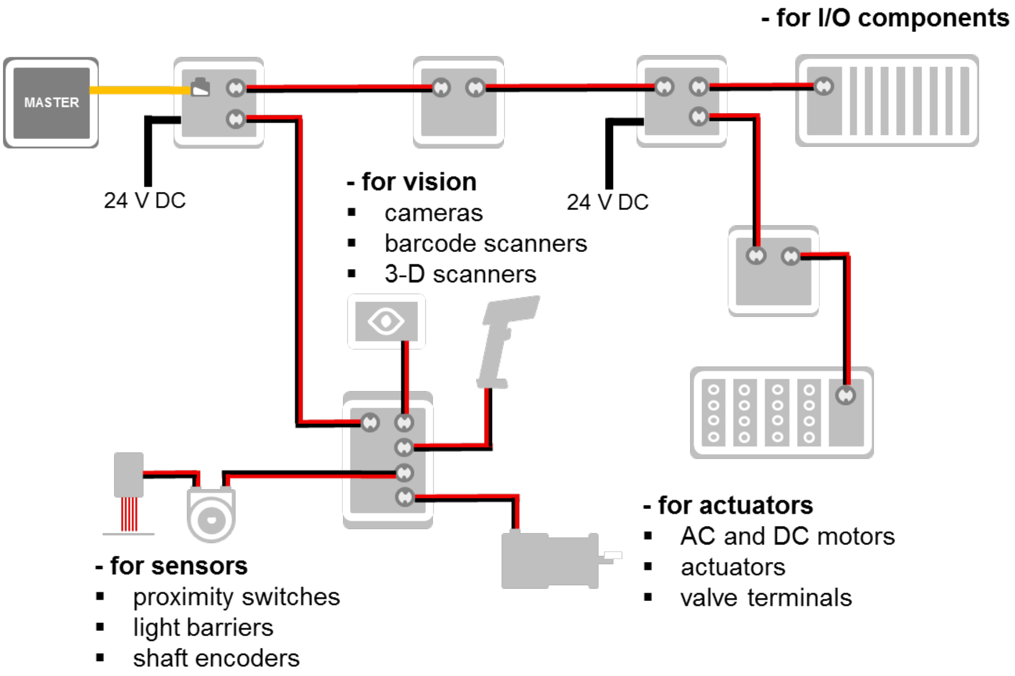


Figure 35: EtherCAT P suites for any kind of devices

1.5. EtherCAT P Device Structure

The ISO/OSI layer model structures communication stacks and specification in the way shown in Figure 36. Taking a reference to EtherCAT, the EtherCAT P functionality and specification is included by the Physical Layer and its specification parts.

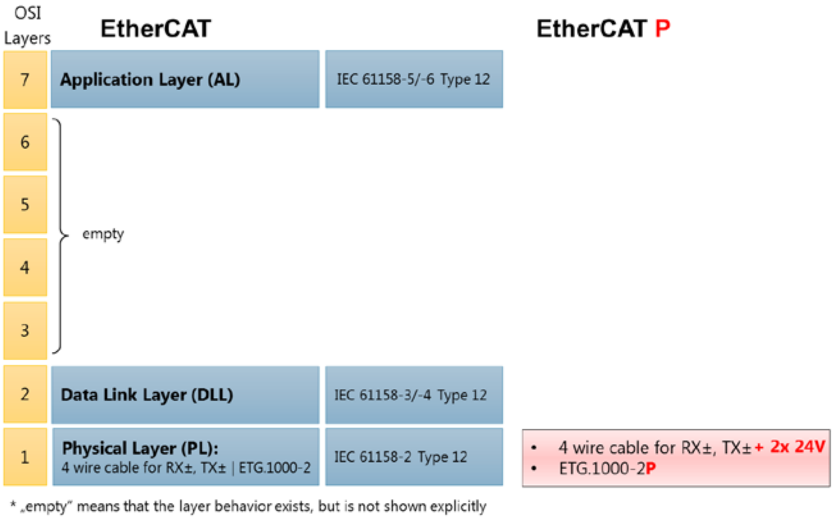


Figure 36: EtherCAT P features specified on PL

EtherCAT P ports can be on the IN port and on one or several OUT ports of an EtherCAT slave.

Figure 37 shows the EtherCAT device structure itself (in green) and the EtherCAT slave's context within the EtherCAT network the relation with the EtherCAT configuration tool.

The EtherCAT slave uses a Standard Ethernet Physical Layer layout to interface to the EtherCAT network. The ESI file describes the EtherCAT features in an xml file. This is provided to the EtherCAT network configuration tool. The configuration tool is used to configure the network layout including a description of the network initialization commands and the cyclic commands. This description is provided to the EtherCAT master using the ENI file.

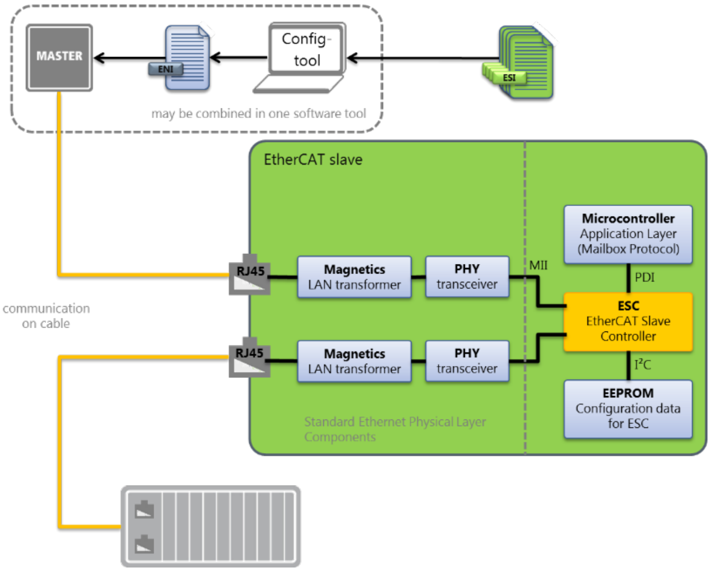


Figure 37: A typical EtherCAT device structure

The add-ons that makes an EtherCAT device to be an EtherCAT P device are shown in Figure 38.

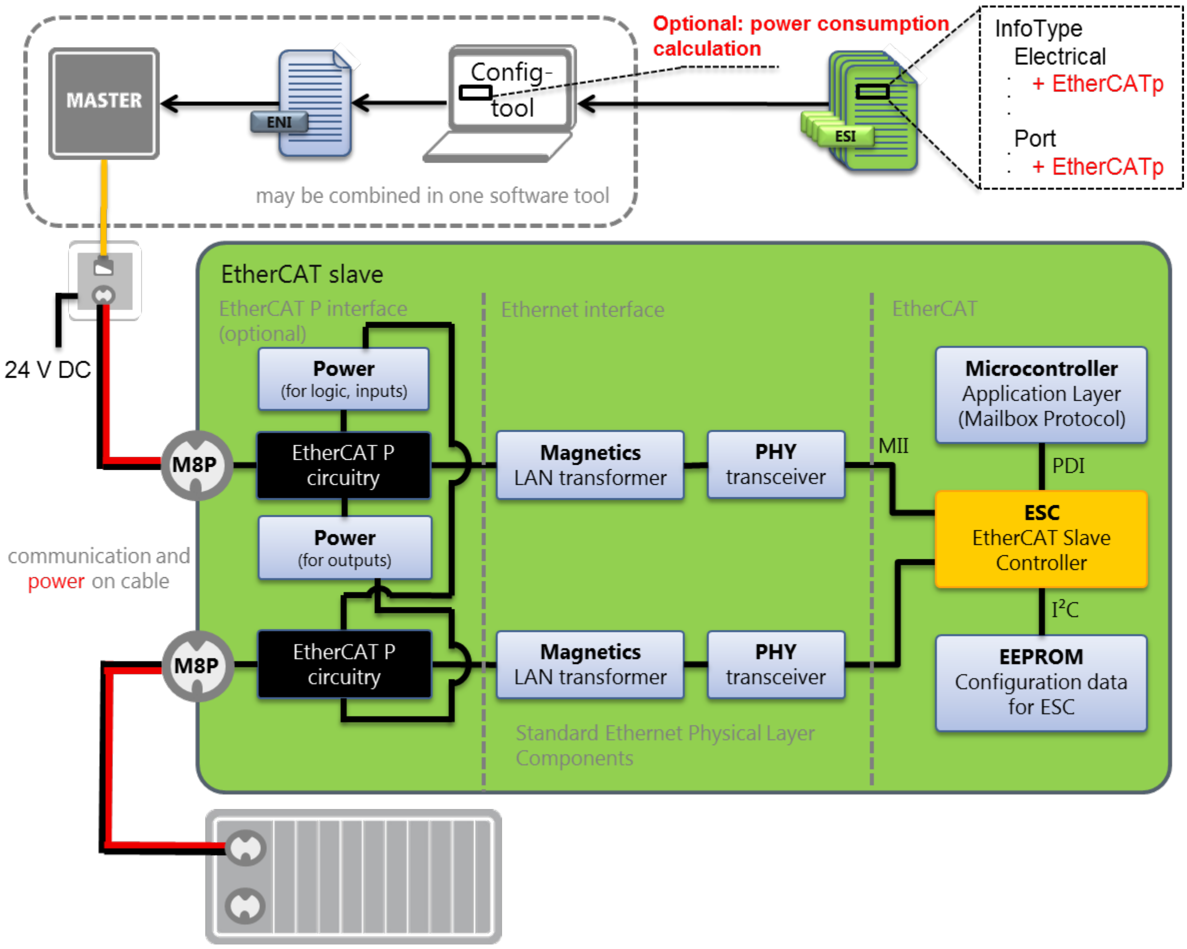

MASTER
ENI
Config-tool
may be combined in one software tool
Optional: power consumption calculation
ESI
InfoType
Electrical
+ EtherCATp
Port
+ EtherCATp
24 V DC
communication and power on cable
EtherCAT slave
EtherCAT P interface (optional)
Ethernet interface
EtherCAT
Power (for logic, inputs)
EtherCAT P circuitry
Power (for outputs)
M8P
M8P
Magnetics LAN transformer
PHY transceiver
MII
Microcontroller Application Layer (Mailbox Protocol)
PDI
ESC EtherCAT Slave Controller
I²C
EEPROM Configuration data for ESC
Standard Ethernet Physical Layer Components

Figure 38: A typical EtherCAT P device structure

The EtherCAT P features on slave side are:

- EtherCAT P interface with the EtherCAT P circuitry and EtherCAT P connector M8 P-coded
- ESI file enhancements to describe the power consumption of the EtherCAT P slave and the power supplied to external sensors/actuators

EtherCAT P networks can be configured with existing configurations tools already – no change on them is required. However, to simplify planning, the configuration tools may be enhanced to calculate and asses the power consumptions in the EtherCAT P segments.

There are no EtherCAT P requirements on the master to make the system work, or, in other words: any existing master can be used to control an EtherCAT P network.

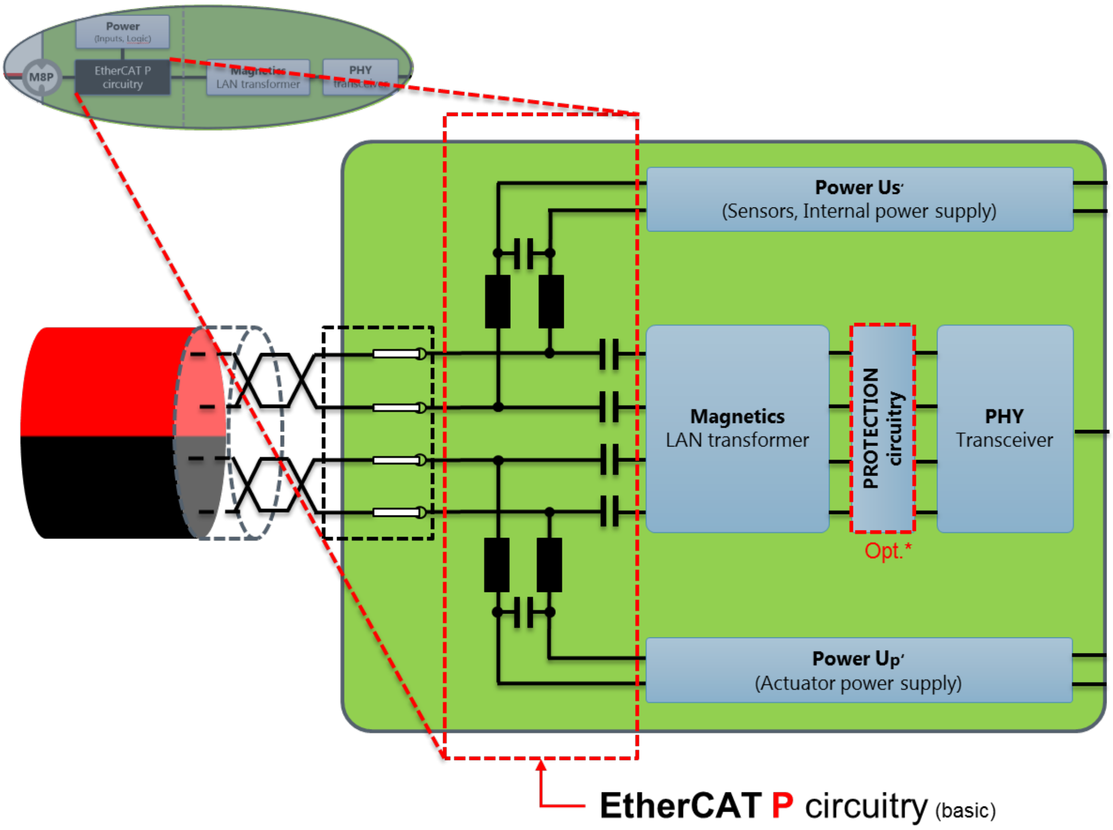


Figure 39: Basic EtherCAT P block diagram for circuitry on the IN port

The four wires are RX± and TX±, which also carry Us$_{24V/GND}$ and Up$_{24V/GND}$. They are connected to the IN port.

The capacitors between the EtherCAT P connector and the Magnetics describe a **high pass filter**: They are transparent for the high frequencies, but block the DC currents of Us and Up.

The LC combination describes a ***low pass filter***: They pass-through the DC currents of Us/Up with 24V each, but block the high frequencies of the communication signals.

1.5.1. EtherCAT P in the ESI file

The ESI file describes (Figure 40) that the EtherCAT slave is actually an EtherCAT P slave (either as Power Sourcing Device or Powered Device), and what its power-consumption characteristics are.

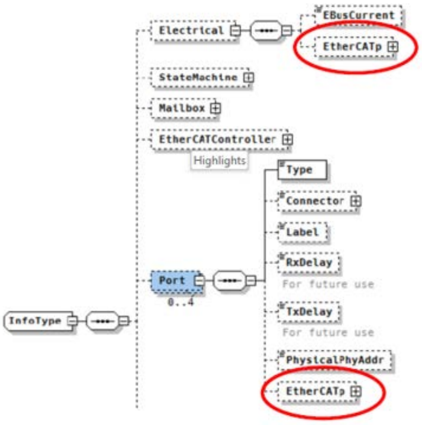


Figure 40: Description of EtherCAT P power consumption characteristics in the ESI file

1.5.2. EtherCAT P Device Types

EtherCAT P distinguishes the following three EtherCAT P device types to describe if they are consuming power or supplying power to the EtherCAT P system.

- Powered Device (PD):
  Uses the power supplied on its IN port
- Power Sourcing Device (PSD):
  The electronics of the device itself and the power supplied to all OUT ports is taken from an **external** power supplied to the PSD. Power supplied via the IN port is not forwarded to the OUT ports nor used by the PSD itself.
- Passive Device

1.5.3. EtherCAT P Device Categories

Figure 41 shows a mixed EtherCAT / EtherCAT P network with the different EtherCAT P device categories.

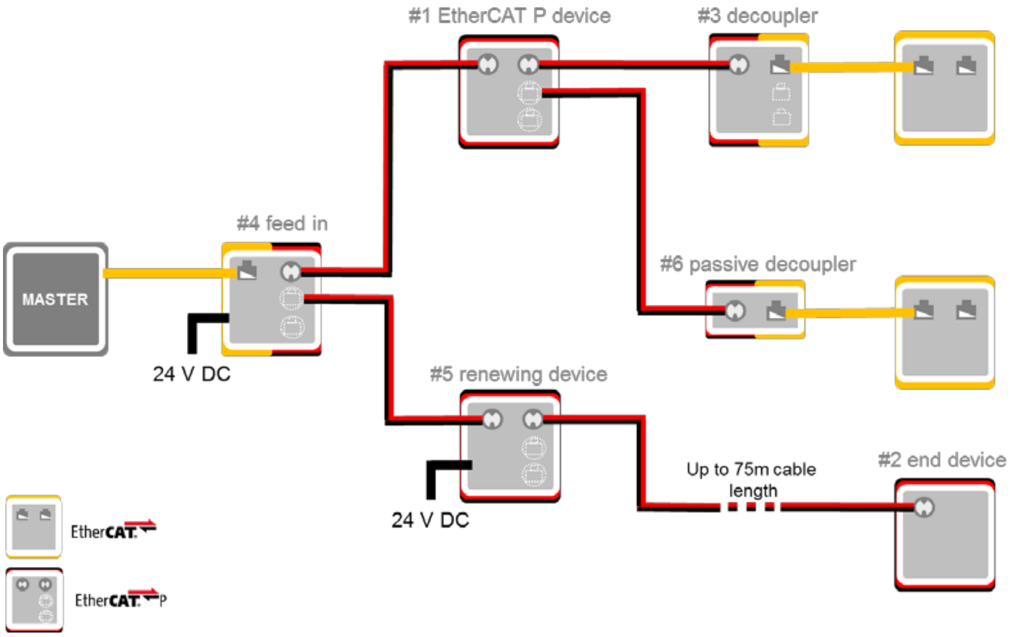


Figure 41: Mixed EtherCAT / EtherCAT P network with all EtherCAT P device categories

Table 17 describes the different EtherCAT P device categories.

Table 18: EtherCAT P Type, Category and Description

#	Type	Category	Description
1	PD	device	“Standard” EtherCAT P device with an EtherCAT P IN port and at least one EtherCAT P out port. Other OUT ports may also be EtherCAT.
2	PD	end device	Only an EtherCAT P IN port. Suits ideally for very small EtherCAT devices (e.g. proximity sensors)
3	PD	decoupler	Decouples power and EtherCAT on the EtherCAT P IN port and has EtherCAT OUT ports. This device has an ESC inside.
4	PSD	feed in	Builds the start of the EtherCAT P segment on the OUT port, while it combines the EtherCAT signal from the IN port and Us/Up (taken from **external** power supply)
5	PSD	renewing device	EtherCAT P device using **external** power supply to refresh Us and Up on the OUT ports. The power supplied on the EtherCAT P IN port is not used any more.
6	Passive	passive decoupler	A decoupler just without ESC (this is the only EtherCAT P device without ESC)

2. EtherCAT P Specification and Documents

EtherCAT P has been included into the EtherCAT standards. EtherCAT has been described by means of the IOS/OSI layer models. Hence, the EtherCAT P specifications can be navigates in the same manner.

Table 19 provides an overview of EtherCAT P related specifications and documents. This also includes the Application Note serving as a practical EtherCAT P Implementation Guide on a doing level – also marked as “top reading” below.

Table 19: EtherCAT P Information, Standards and References

Introduction	Articles	EtherCAT P has been introduced in several articles. A selection of them is given here. → PC Control (English): 01/2016 | (German): 01/2016 (stronger technical footprint) → PC Control (English): 01/2016 | (German): 01/2016 (system view)
	Proceedings of ETG Events	Minutes of the Technical Committee Meetings give additional background information. EtherCAT P was introduced on the spring meeting 2016. Meeting minutes from then on are of specific EtherCAT P interest. → www.ethercat.org → Downloads → Select Filter: Proceedings and Papers → Technical Committee Meeting
Development	Communication Slides	The communication slides provide a broad description of EtherCAT mechanisms for developers. It also describes some basics on the Physical Layer, which, naturally, include some EtherCAT P basics, too. → English
	Application Note TOP READING	While the EtherCAT specifications describe the EtherCAT P technology in a more formal context, the Application Note aims to give very practical guidance on a doing-level. It includes details for both, PSD and PD. This includes EtherCAT P schematic details, electronic components, layout recommendations for grounding/ EMI/ EMC/ layout and examples. → AN ECATP Implementation Guide www.ethercat.org/ethercatp
Specifications	EtherCAT P specification	The main EtherCAT P specification document. It describes: Voltages, System Architecture, Device Types, Powered Devices, Power Sourcing Devices, Passive components, Device Categories, Physical Layer Extension, Cables, Connectors. It refers to related specifications → ETG.1030 www.ethercat.org/ethercatp
	EtherCAT P Connector	Specification of M8 P-coded connector. Any cable manufacturer can produce and sell such connector. The M8 P-coded connector has also been submitted for IEC standardization. → ETG.1030.1 www.ethercat.org/ethercatp
	EtherCAT P Physical Layer extension	EtherCAT P physical layer specifics. → ETG.1000.2 P www.ethercat.org/ethercatp
	EtherCAT Slave Information (ESI)	Description of EtherCAT P specific details in the ESI file, such as EtherCAT P device type (PD, PSE), min/max voltages and load types. The EtherCAT P part of the ESI specification is currently maintained in a separate document (named EtherCAT Slave Information (EtherCAT P). → ETG.2000 (ECATP): www.ethercat.org/ethercatp The related schema file is also available for download → EtherCATInfo.xsd (and related xsd files): www.ethercat.org/ethercatp
	ETG.9001 Marking Rules	As with EtherCAT and Safety over EtherCAT, for EtherCAT P a logo and trademark are defined. This and their usage are specified in ETG.9001. The EtherCAT P part of the ESI specification is currently maintained in a separate document (named EtherCAT Marking Rules (EtherCAT P). → ETG.9001 (ECATP): www.ethercat.org/ethercatp

3. EtherCAT P Conformance Testing

3.1. General

Since EtherCAT P combines power and data on the same cable, a faulty implementation might influence the whole system. Furthermore, wrong power consumption entries in the ESI file may lead to network configurations that do not work reliably, since the planning tool relies on this data.

Therefore, for EtherCAT P devices the Physical Layer Test is mandatory. EtherCAT P enhancement in the ESI file are tested with the default test set included in the CTT.

In the introductory phase the EtherCAT P physical layer test is available in Germany ***-free of charge*** – retesting as well!

3.2. Evaluate your current EtherCAT Slave for EtherCAT P

To evaluate your current slave’s EtherCAT interface to be used as basis for an EtherCAT P interface PCB, ETC Germany provides a set of test adaptors. It is sufficient to make a first pragmatic test on specific operating situation of the PHY: The CTT Test File executed in the set-up shown in Figure 42 allows to exclude a communication backdraw caused by the Base Line Wander (BLW) effect. It will also check for communication issues induced by the power supplies.

The Test requires a CU2501, 2 EtherCAT P test adapter and the CTT. The Test File TF-1000 checks the functionality, and is used with three different cable lengths. The figure below shows the testing setup with a 75m long cable.

To borrow such an set of adaptors contact ETC Nuremberg (etc@beckhoff).

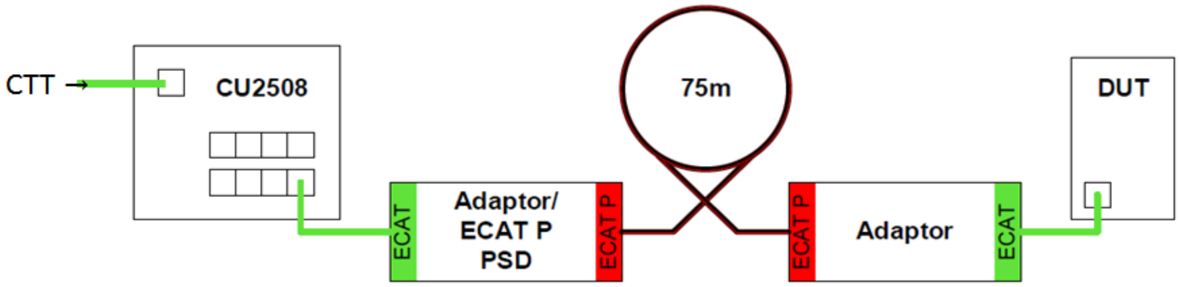


Figure 42 Baseline wander testing with a 75m cable

3.3. Contact for EtherCAT P Conformance Testing

As with EtherCAT and Safety over EtherCAT, contact conformance@ethercat.org for EtherCAT P conformance testing.

4. EtherCAT P Licensing

4.1. General

Like EtherCAT, EtherCAT P is a protected technology – this helps to ensure compatibility and interoperability. This concept has proven to be very successful with EtherCAT and therefore is also applied to EtherCAT P: Implementing EtherCAT P in products requires a license. Again, as with EtherCAT itself, Beckhoff as inventor of EtherCAT P is supporting and encouraging the wide spread adoption of EtherCAT P. Therefore, the license for EtherCAT P is ***free of charge***. For interoperability reasons, EtherCAT P may only be used with specified connectors.

The hybrid connectors will be licensed separately (to connector makers). Users and device vendors do not need an additional license for using them.

4.2. License Agreement

EtherCAT P licensing is particular simple if you have already signed an “EtherCAT Technology Family License Agreement” with Beckhoff - Beckhoff then provides a side letter. Newly issued License Agreements already include EtherCAT P.

Contact licensing@beckhoff.com regarding EtherCAT P.

5. How to Develop an EtherCAT P Slave

The previous chapters have provided all the basic insight and references to implement an EtherCAT P device. Of course, all EtherCAT implementation related steps remain. Sections I and Section II provide comprehensive information on it.

Regarding the EtherCAT P specific part, no matter if started with a new device from scratch or enabling an existing EtherCAT device with EtherCAT P, the implementation of the EtherCAT P specific part goes along the following few steps:

1. ETG membership
2. License Agreement
3. Study Application Note and EtherCAT P specifications
4. Design EtherCAT P specific PCB along Application Note
5. Use already available EtherCAT P devices for pragmatic functionality testing
6. To configure the test network, use an EtherCAT Configuration Tool supporting the configuration of EtherCAT P networks
7. Update/use label and trademark term (Figure 44)
8. Contact conformance@ethercat.org for EtherCAT P Conformance testing (optionally, and recommended, also for EtherCAT Conformance testing

Figure 43: EtherCAT P logo

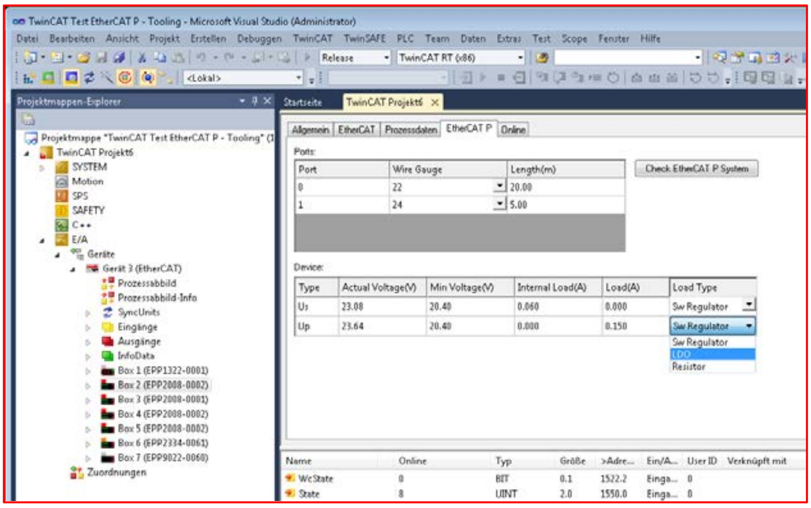
5.1. EtherCAT P and EtherCAT Configuration Tool

The EtherCAT configuration tools task is to generate a network description, standardized as EtherCAT Network Information (ENI) within ETG. It describes the topology, all EtherCAT slaves with their assigned EtherCAT address, the initialization commands for each slave and the cyclic commands to exchange cyclic input and output data between master and slaves. All this remains unchanged. In fact, no change at all is necessary on the EtherCAT configuration tool to run EtherCAT P slaves in a network. Of course, as any fieldbus slave, power-supply must be guaranteed.

As mentioned earlier already, for simplify planning, the configuration tools may include functionality to calculate and asses the power consumptions in the EtherCAT P segments. It verifies if the daisy-chained power is sufficient for each individual EtherCAT P slave and its connected loads.

Once the EtherCAT power supplies calculations have been finalized, the actual EtherCAT network configuration as described above can be done and the master can run the EtherCAT / EtherCAT P network without even knowing of EtherCAT P details.

Figure 44 shows how an EtherCAT P planning tool is integrated into the EtherCAT network configuration tool to verify the power supply of the EtherCAT P segments.



Ports:

Port	Wire Gauge	Length(m)
0	22	20.00
1	24	5.00

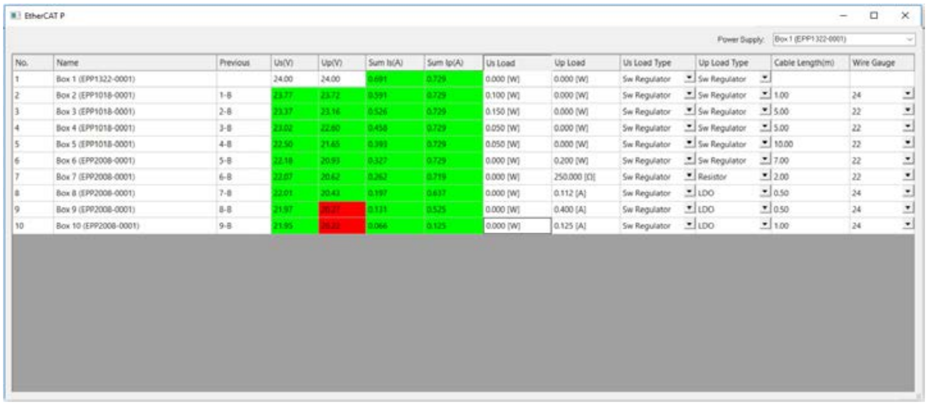
Device:

Type	Actual Voltage(V)	Min Voltage(V)	Internal Load(A)	Load(A)	Load Type
Us	23.08	20.40	0.060	0.000	Sw Regulator
Up	23.64	20.40	0.000	0.150	Sw Regulator

Name	Online	Typ	Größe	>Adre...	Ein/A...	User ID	Verknüpft mit
WcState	0	BIT	0.1	1522.2	Einga...	0	
State	8	UINT	2.0	1550.0	Einga...	0	

Figure 44: EtherCAT P planning tool integrated into a network configuration tool (Beckhoff)

A table shows if the power supply meets the power consumption of each individual EtherCAT P slave in (Figure 45). Also, the load and load types of each EtherCAT P slave can be configured.



Power Supply: Box 1 (EPP1322-0001)

No.	Name	Previous	Us(V)	Up(V)	Sum Is(A)	Sum Ip(A)	Us Load	Up Load	Us Load Type	Up Load Type	Cable Length(m)	Wire Gauge
1	Box 1 (EPP1322-0001)		24.00	24.00	0.691	0.729	0.000 [W]	0.000 [W]	Sw Regulator	Sw Regulator		
2	Box 2 (EPP1018-0001)	1-B	23.77	23.72	0.591	0.729	0.100 [W]	0.000 [W]	Sw Regulator	Sw Regulator	1.00	24
3	Box 3 (EPP1018-0001)	2-B	23.37	23.16	0.526	0.729	0.150 [W]	0.000 [W]	Sw Regulator	Sw Regulator	5.00	22
4	Box 4 (EPP1018-0001)	3-B	23.02	22.60	0.458	0.729	0.050 [W]	0.000 [W]	Sw Regulator	Sw Regulator	5.00	22
5	Box 5 (EPP1018-0001)	4-B	22.50	21.65	0.393	0.729	0.050 [W]	0.000 [W]	Sw Regulator	Sw Regulator	10.00	22
6	Box 6 (EPP2008-0001)	5-B	22.18	20.93	0.327	0.729	0.000 [W]	0.200 [W]	Sw Regulator	Sw Regulator	7.00	22
7	Box 7 (EPP2008-0001)	6-B	22.07	20.62	0.262	0.719	0.000 [W]	250.000 [Ω]	Sw Regulator	Resistor	2.00	22
8	Box 8 (EPP2008-0001)	7-B	22.01	20.43	0.197	0.637	0.000 [W]	0.112 [A]	Sw Regulator	LDO	0.50	24
9	Box 9 (EPP2008-0001)	8-B	21.97	20.27	0.131	0.525	0.000 [W]	0.400 [A]	Sw Regulator	LDO	0.50	24
10	Box 10 (EPP2008-0001)	9-B	21.95	20.22	0.066	0.125	0.000 [W]	0.125 [A]	Sw Regulator	LDO	1.00	24

Figure 45: Showing if power-supply is sufficient or not

6. EtherCAT P Development Support

As with EtherCAT, the EtherCAT Technology Group supports you with your EtherCAT P implementation, contact info@ethercat.org.