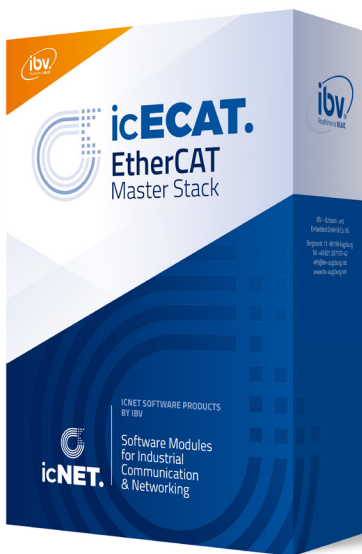




icECAT. EtherCAT Master Stack for Embedded Systems

TECHNOLOGY **EtherCAT**



CHARACTERISTICS

- EtherCAT Master stack, especially designed for use on embedded systems: best performance with low resource usage
- Target platforms: small microcontrollers up to Industrial PCs
- Written portable in ANSI-C. Operating system and network interface dependent parts are located in separate modules to allow a simple adaption to new platforms
- Different options for Ethernet communication interface:
 - Raw Ethernet access via network driver of the operating system (e. g. Raw Socket, BPF, PCAP)
 - Optimized link layer driver with zero-copy buffers for shorter cycle times and reduced CPU load
- Shipped in source code, project-based license, no royalties

ARCHITECTURE

- The master stack is a passive library which provides a C API to the application. The cyclic and acyclic tasks can be driven by the application. Different approaches for scheduling (single threaded, multi-threaded) and cycle synchronization are possible.
- Easy integration in applications: The stack does not impose the software architecture to the application. Multiple instances of the master stack are possible in one application.
- Scalable architecture: Features of the stack can be disabled at compile-time to minimize the footprint.

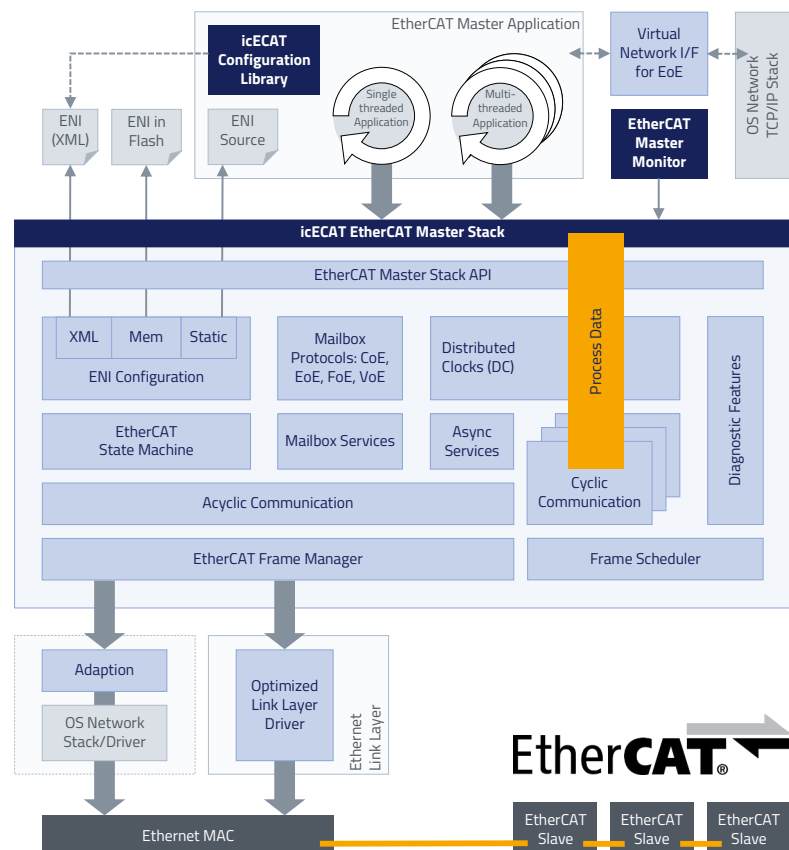
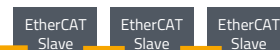


Figure 1: Architecture of the icECAT EtherCAT Master Stack and adaption to the EtherCAT master application. Access to Ethernet MAC via the network stack of the operating system (left side) or an optimized link layer driver (right side).

EtherCAT



FEATURES

Basic features

- EtherCAT master stack, compliant to ETG.1500 EtherCAT Master Class-B feature set and most important Class-A features
- Support of all EtherCAT (DLPDU) datagrams [101](#)
- Support of communication with simple and complex slave devices [103](#)
- Checking of the working counters, slave responses and lost frames [105](#)
- Enhanced error handling of slave and network errors
- Support of EtherCAT frames over Ethernet as link layer [107](#)

EtherCAT network configuration

- EtherCAT network configuration via ENI file [301](#)
- Integrated XML parser, independent of the operating system. Access to ENI information via ANSI C file operations
- Alternative network configuration: Static ENI information in source code which can be generated from an ENI file. No file system is necessary.
- Comparison of configured and existing network configuration (via ENI configuration) [302](#)
- Reading out of ESC EEPROM (via ENI configuration) [305](#)
- API to read out the network configuration

EtherCAT state machine (ESM)

- Support of the EtherCAT State Machine for each configured slave [104](#)
- Parallel initialization of multiple EtherCAT slaves
- API to control the state of the EtherCAT master and each EtherCAT slave
- State monitoring of the EtherCAT slaves

Cyclic communication (PDO)

- Support of process data exchange in cyclic frames [201](#)
- Large process images over multiple frames are possible
- API to access to the process image either raw or via I/O variables as defined in the ENI file. Bitwise access to I/O variables and big-endian conversion are supported.
- Optimized for speed: Outgoing and incoming process image can be directly accessed in the cyclic frame without copying the data.
- Macros for high performance access to the process data
- Support of one or more cyclic tasks with different process images and different cycle times [202](#)
- Cyclic tasks are driven by the application. The application can e. g. use a hardware specific timer or it can synchronize the task to other external events.

Acyclic communication

API to send asynchronous frames with EtherCAT datagrams by the application

Slave-to-Slave communication

Slave-to-Slave communication for PDOs is handled by the master [1201](#) (used e. g. for communication between an FSoE master and an FSoE slaves).

Frame scheduling

Different modes for scheduling of cyclic and acyclic frames:

- IMMEDIATE: Frames are sent by each task without scheduling
- CHAINING: Frames are sent by one task in the priority defined in the ENI. The frame of the acyclic task can be chained to one cyclic task.
- SCHEDULED: Frames can be sent by a separate scheduler thread (see scheduling sample in figure 2)

Mailbox support

Support of EtherCAT mailbox transfers [401](#) [402](#). Mailbox communication can get handled polled [404](#) or synchronized with mailbox status events from the cyclic frames.

CAN application layer over EtherCAT (CoE)

- Support of SDO upload and download (normal and expedited transfers), segmented transfers and entire objects at once [501](#) [502](#) [503](#)
- Supporting emergency messages is possible in the EtherCAT application [505](#)
- API with non-blocking and blocking functions to CoE functions

Ethernet over EtherCAT (EoE)

- Support of tunneling Ethernet communication in the acyclic frame [601](#)
- Support of a virtual switch functionality [602](#)
- Interface to the network stack of the host operating system (only for some supported OS) [603](#)
- Frame mirroring support to monitor the cross-slave traffic

File transfer over EtherCAT (FoE)

- Support of file transfer to/from slaves (no local file system is necessary) [701](#)
- Support of firmware up-/download with password and filename [702](#)
- Support of Bootstrap state for slave firmware update [703](#)

Vendor specific over EtherCAT (VoE)

Support of vendor specific protocol with direct access to mailbox protocol [1001](#)

FEATURES

Distributed Clocks (DC)

- Support of Distributed Clocks in master mode.
The master synchronizes the network to the DC clock of a reference slave [1101](#).
- Support of DC slave mode (with limitations)
- Continuous monitoring of the synchronization difference in the slaves [1103](#)
- Runtime monitoring of the DC accuracy

Cable Redundancy

- Support of the FP Cable Redundancy as compile configuration: Basic functions [FPCR_101](#) and diagnosis functions [FPCR_102](#)

Diagnostic features

- Application event API for error notification with detailed error information
- Various statistics and error counters
- Developer logging for different software layers

Features on the roadmap

- Support of SDO information services to read out the slave's object dictionary [504](#)
- Access to Slave Information Interface (SII), reading and writing slave EEPROM
- Explicit device identification and station alias addressing [303](#) [304](#)

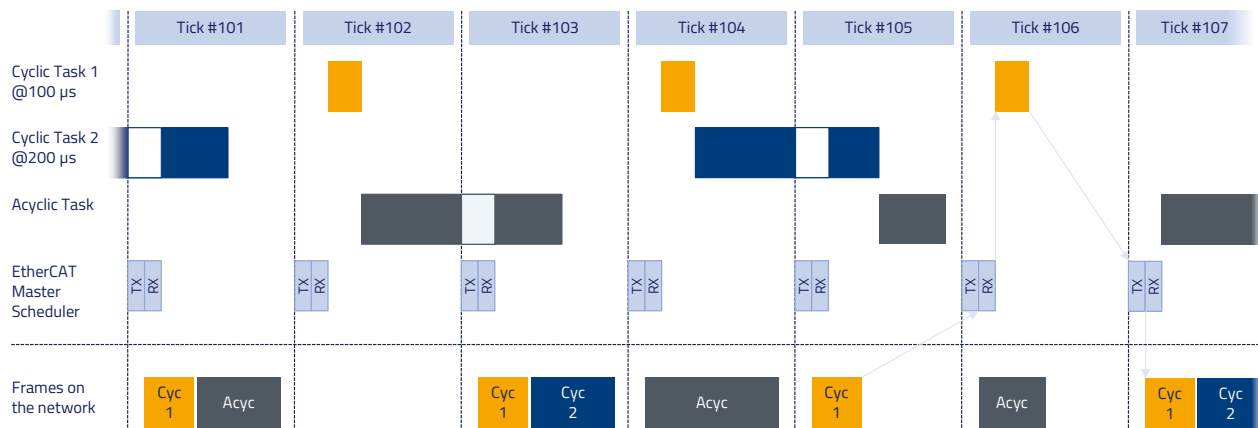


Figure 2: Scheduling example with optimal usage of the EtherCAT network bandwidth

- Cyclic task 1 is running each 100 µs.
- Cyclic task 2 is running each 200 µs. It needs more processing time than cyclic task 1.
- EtherCAT master scheduler is driven with 50 µs cycle time and runs with highest priority to ensure the real-time communication.
- Acyclic task is running in background with lowest priority.

TOOLING

EtherCAT Master Monitor

Command line tool for visualizing the status of the EtherCAT network and controlling the EtherCAT master and slaves. The tool can be started at run-time on the master's target or on a remote host (via a TCP connection). It supports:

- List and control state of the master and all slaves
- View and modify I/O variables of the process images
- Show statistic and error counters
- Manual access to ESC slave registers
- Manual sending of SDO commands

ENI Tool

Command line tool for converting ENI files into source code

EtherCAT Configuration Library

see also: www.ibv-augsburg.de/icnet/ethercat-configuration

Available as separate product:

- EtherCAT configuration library for integration in an own configuration application or engineering tool. Written in ANSI-C
- Divisible in server part (running on target) and client part (running on configuration system)
- Online network scanning on target system [301](#)
- Slave information is taken from ESI files
- Generation of an ENI file



SUPPORTED PLATFORMS

Operating System:

Linux with or without Preempt-RT Patch
Xenomai v2.6, v3^{*)}
QNX Neutrino RTOS v6.x, v7
Texas Instruments TI-RTOS^{*)} (coming soon)
FreeRTOS^{*)}
„Bare-Metal“, no operating system^{*)}

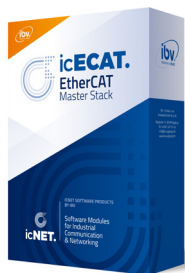
CPU Architecture:

x86, x64, ARM
x86, x64, ARM
x86, x64, ARM, PPC
ARM Cortex-A, ARM Cortex-R
ARM Cortex-M, ARM Cortex-A
ARM

^{*)} optimized link layer driver is necessary

AVAILABLE OPTIMIZED LINK LAYER DRIVERS

- Intel I210
- Intel I225
- Texas Instruments Sitara (coming soon)
- NXP i.MX 6, i.MX 8
- NXP i.MX RT1064, i.MX RT1050
NXP LPC 18xx
- STMicroelectronics STM32:
STM32H7, STM32F7, STM32F4
- others on request



LICENSING

The **icECAT. EtherCAT Master Stack** is offered under a project-based license for a one-time fee without royalties. The libraries and tools are shipped in source code.

SHIPMENT

icECAT. EtherCAT Master Stack consists of:

- EtherCAT Master Stack and Monitor Tool (in source code)
- Programmer's Guide
- Reference Manual
- Integration Guide for microcontroller targets
- Sample applications
- One Standard Link Layer Driver
- One day integration support

EVALUATION VERSION

An evaluation of the **icECAT. EtherCAT Master Stack** is possible on the following platforms:

- Linux Preempt-RT
- QNX
- STM32H7 on NUCLEO-H743ZI2 development board
- NXP i.MX RT1064 on NXP i.MX RT1064 EVK board
- other platforms on request

Support and Services

IBV provides professional engineering, integration, support and consulting services for Industrial Ethernet technologies on real-time and embedded systems.

Contact

For getting more details or for requesting an evaluation version, please visit our website or write an email

www.ibv-augsburg.de/icnet/ethercat-master

icecat@ibv-augsburg.de

About IBV

Custom software solutions for real-time and embedded systems. The company shows a broad expertise in embedded systems and real-time applications. The services and products made by IBV are used in the industrial automation, medical devices, IoT and smart home applications, telecommunication, test and measurement. In addition to the engineering services, IBV offers software products for implementing Industrial Ethernet solutions.

IBV - Echtzeit- und Embedded GmbH & Co. KG
Bergiusstraße 13
D-86199 Augsburg
Germany
Tel. +49 821 207107-42
www.ibv-augsburg.de

ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany. QNX® is a registered trademark of QNX Software Systems.