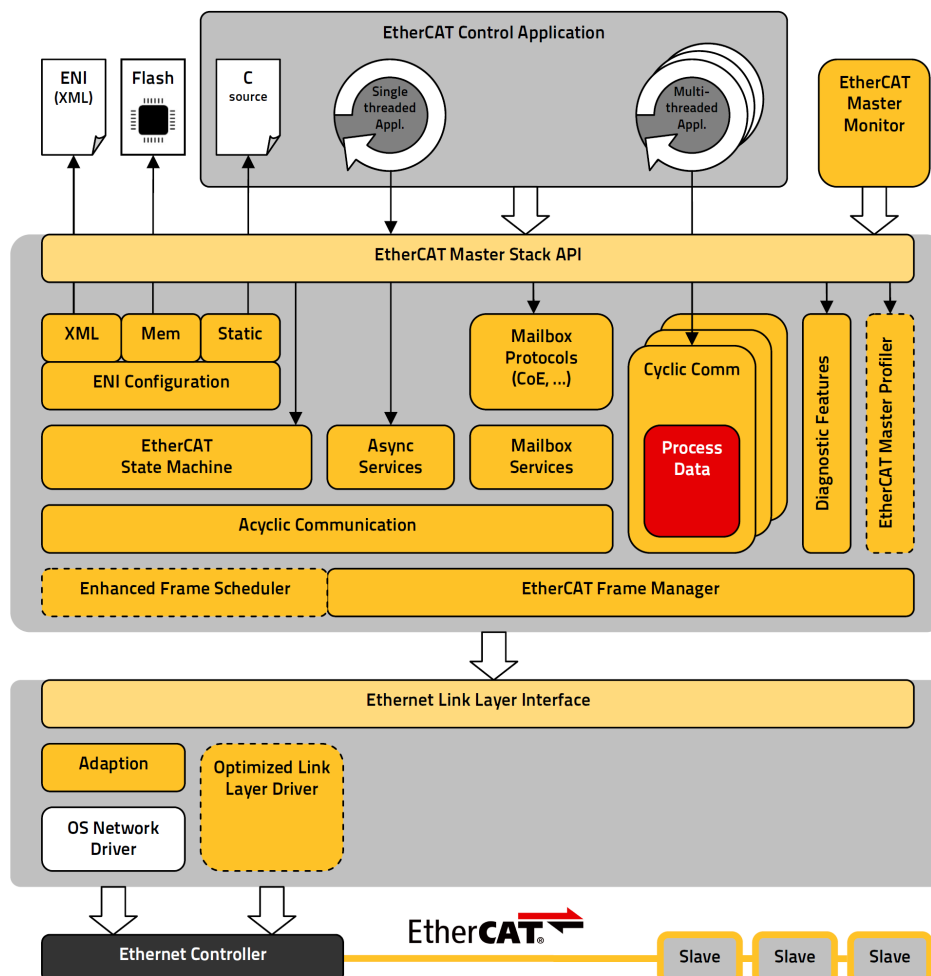


# icECAT. EtherCAT Master Stack for Embedded Systems

## Characteristics

- EtherCAT Master stack, specifically designed for embedded systems (best performance with low resource usage)
- Different options for Ethernet communication interface:
  - Raw Ethernet access via network driver of the operating system (e.g. BPF, Raw Socket, NDIS)
  - Optimized link layer driver with zero-copy buffers for shorter cycle time and reduced CPU load
- Real-time capability (dependent on underlying operating system)
- Scalable architecture: Features of the stack can be enabled resp. disabled at compile-time to reduce code size.
- Written portable in ANSI-C, little-endian and big-endian architectures are supported. Operating system and network interface dependent parts are located in separate modules to allow a simple adaption to new platforms.
- Easy integration in applications. The stack does not impose the software architecture to the application. Different approaches for scheduling (single threaded, multi-threaded) and cycle synchronisation are possible.
- Adaption with **EMBRICS**<sup>®</sup> software module **ioECAT** possible. This software layer above allows an access to logical I/O objects independent of the underlying EtherCAT protocol.
- Shipped in source code, project based license, no royalties

## Architecture



## Features

### = EtherCAT Feature ID according to ETG.1500

### Basic features

- EtherCAT Master Stack, compliant to ETG.1500 EtherCAT Master Class-B feature set
- Multiple EtherCAT master stack instances on different network adapters are possible
- 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 [V1.x]
- 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 or via callouts to access any type of memory (e.g. flash)
- Online network scanning as alternative network configuration [V1.1] **301**
- Further alternative network configuration: Static ENI information in source code which can be generated from an ENI file (no file access necessary) [V1.1]

- 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**
- State transition dependent initialization of the EtherCAT slaves according to the ENI configuration
- 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 with each cyclic frame (via ENI configuration)

### Cyclic communication (PDO)

- Support of process data exchange in cyclic frames **201**
- Large process images over multiple frames are possible
- Support of one or more cyclic tasks with different process images and different cycle times [V1.1] **202**

- Cyclic communication can be done in separated threads for each process image or in one thread together with the acyclic communication.
- 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.
- Configurable cycle time supervision
- 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
- Support of frame repetition to increase communication robustness [V1.x] **203**

### Acyclic communication

- API to send asynchronous frames with EtherCAT datagrams by the application

### Option: Enhanced frame scheduler

- The scheduler helps to optimize the usage of the network bandwidth for acyclic frames with respecting real-time requirements of the cyclic frames [V1.1]

## Mailbox support

- Support of EtherCAT mailbox transfers [401](#) [402](#)
- Mailbox communication can get handled polled [404](#). Optionally, it be synchronized with mailbox status events from the cyclic frames to improve the performance and reduce network load

## CAN application layer over EtherCAT (CoE)

- Support of SDO upload and download (normal and expedited transfers) [501](#), segmented transfers [502](#) and entire objects at once [503](#)
- Automatic slave configuration via CoE (via ENI configuration)
- Support of SDO information services to read out the slaves object dictionary [V1.1] [504](#)
- Support of emergency messages possible in the EtherCAT application [505](#)
- API with non-blocking and blocking functions to CoE functions

## Slave-to-Slave communication

- Slave-to-Slave communication via ENI configuration handled by master [1201](#)

## Diagnostic features

- Application event API for error notification with detailed error information
- Various statistics and error counters

- Support of virtual variables in the cyclic communication with diagnosis information (as configured in ENI file) [V1.1]
- Developer logging for different software layers (if enabled in compile configuration)

## Features on the roadmap

- Support of Distributed clocks [V1.1]
- Support of Ethernet over EtherCAT (EoE) mailbox protocol
- Support of File Transfer over EtherCAT (FoE) mailbox protocol
- Access to Slave Information Interface (SII), reading and writing slave EEPROM
- Support of cable redundancy
- Support of hot connect, explicit device identification and station alias addressing [303](#) [304](#)
- Support of IRQ field in datagram header [102](#)

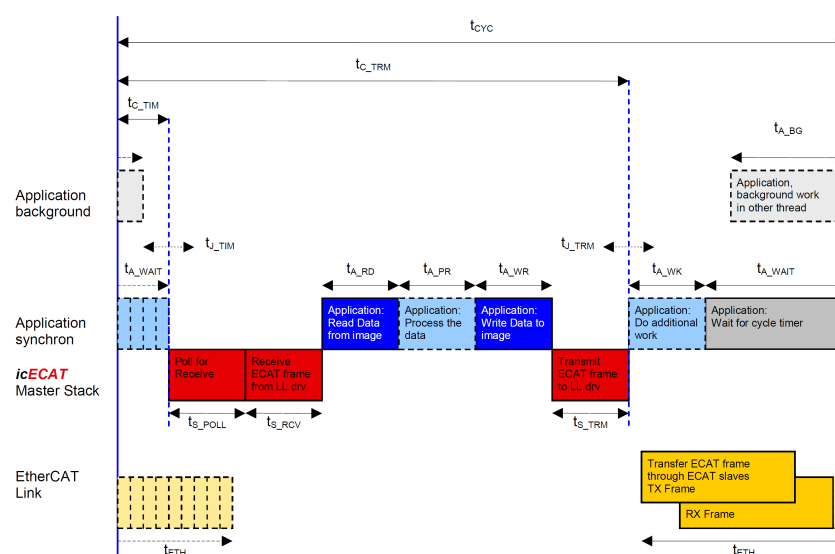
## Tooling

### 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 send of SDO commands

### EtherCAT Master Profiler

- Tool for measuring the performance of cyclic frames on a custom hardware and a custom EtherCAT network
- Early evaluation of the hardware and software platform on different link layers is possible



## Supported Platforms

### Operating System:

- QNX Neutrino RTOS
- Linux Kernel with or without RT Preempt Patch
- Xenomai
- Microsoft Windows 7 [V1.1]
- Microsoft Windows Embedded Compact 7 / 2013 [V1.2]
- FreeRTOS (optimized link layer driver necessary) [V1.1]

### CPU Architecture:

- x86, ARM, PPC
- x86, ARM, PPC
- x86, ARM, PPC
- x86
- x86, ARM
- ARM

## Available Optimized Link Layer Drivers

- Texas Instruments Sitara AM335x
- Freescale i.MX6 [V1.1]
- Freescale QorIQ P20xx [V1.1]
- NXP LPC18xx [V1.1]
- others on roadmap

## Licensing

- The **icECAT**. EtherCAT Master Stack is offered under a project based license (without royalties)

## Shipment

**icECAT**. EtherCAT Master Stack consists of:

- EtherCAT Master Stack, Monitor Tool and Profiling Tool (in source code)
- User Manual and Reference Manual
- Sample application (in source code)

## **icECAT**. Slave SDK (TI AM335x) - another Product of the **icECAT**. Family

- Software Development Kit for simple implementation of an EtherCAT slave device on the Texas Instruments Sitara AM335x ARM Processor Family
- **icECAT**. Slave library interacts with the PRU (Programmable Real-time Unit) of the TI AM335x
- The SDK integrates the Beckhoff ET9300 slave stack (source code is available for free to ETG members)
- The SDK contains a guide which helps to integrate the software to build an EtherCAT slave device
- The SDK is available for Linux and Xenomai
- For more details about the **icECAT**. Slave SDK, contact [icecat@ibv-augsburg.net](mailto:icecat@ibv-augsburg.net)
- IBV provides professional services for embedded software development

## About IBV

*IBV - Echtzeit- und Embedded GmbH & Co. KG is located in Koenigsbrunn near Augsburg in Germany. IBV is focusing on the development of software for various technical markets. As competent partner for software development and operating systems, IBV provides "all-in-one" services for embedded projects. For more information visit [www.ibv-augsburg.net](http://www.ibv-augsburg.net)*

ARM and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.  
EtherCAT<sup>®</sup> is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.  
Microsoft<sup>®</sup>, Windows<sup>®</sup> are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.  
QNX<sup>®</sup> is a registered trademark of QNX Software Systems.