EtherCAT Slave Stack

- **Significantly reducing time to market for slave development**
- **Mailbox services CoE, EoE, FoE and VoE supported**
- **Cross-platform API allows Rapid Prototyping**

The EtherCAT® Slave Stack is a software stack for the easy development of embedded slave devices written in ANSI C. It's available as source code optimized for high performance and low resource usage.

With the cross-platform API the parallel development of the embedded device hardware and application software is possible. The EtherCAT Slave Stack supports all necessary protocols and was tested successfully with several EtherCAT Slave Controllers (ESC).

Easy and Fast EtherCAT Slave Device Development

- The EtherCAT Slave Stack provides in combination with an EtherCAT slave controller (ESC) all services for an EtherCAT compliant communication according to IEC 61158 / ETG.1000.
- The source code is written in ANSI-C, optimized for embedded targets with respect to performance and resource usage.
- It is available as binary version for various operating systems or as source code tested on different target architectures (x86, PowerPC™, ARM®) with and without operating system.
- An API based interface provides a clear separation between application and stack which makes a later change to an updated stack revision or different hardware easy.
- All protocol complexity and hardware dependance is hidden. The developer can concentrate on application development which significantly reduces the time to market.
- The stack comes with a comprehensive manual and a sample application which makes a good starting point for the application.

Mailbox Protocols CoE, EoE, FoE and VoE

- CoE includes 'SDO Information Service', 'Segmented SDO Service', dynamic PDO assignment and dynamic PDO configuration.
- EoE is implemented via callback for completely assembled frame (from EtherCAT) and a simple function to inject an Ethernet frame into the EtherCAT slave stack which handles fragmentation etc.
- FoE offers callback for each data chunk etc.
- VoE is implemented by simple callback for each mailbox packet of that type.

Object Dictionary

- Comprehensive support for object dictionary and process data - application just accesses the objects and stack handles almost everything, e.g. updates when they are PDO mapped, automatically.
- Dynamic dictionary, completely changeable during runtime.
- With automatic handling of important entries, such as PDOs and PDO assignment objects.
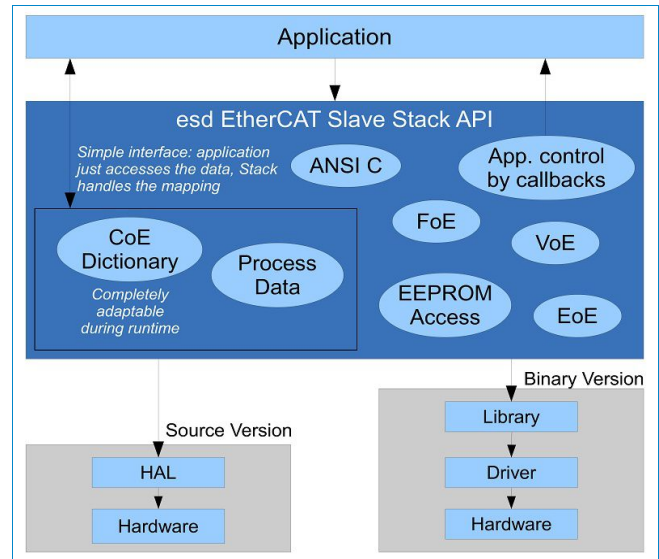- Callbacks for important events, e.g. before and after SDO Download, etc.

Overview

Additional Features

- Includes functions to read/write ESC's EEPROM.
- ESC can be used in polling or IRQ mode.
- Supports multiple ESC by one application.
- Each version of the stack is tested for compliance by using the latest EtherCAT Conformance Test Tool.
- The source code version allows compile time adaption of several features to optimize the resource usage.
- Source code version is based on a well defined Hardware Abstraction Layer (HAL) to adapt the stack to the target hardware with as little effort as possible.

Supported EtherCAT Slave Controllers (ESC)

- ET1100 and compatible ESCs.
- TI AM335X (Sitara™) Programmable Real Time Units.
- Renesas R-IN32M3-EC.
- Supports memory mapped or SPI based I/O.

Rapid Prototyping with esd PCIe Slave Interface

It is possible to start developing the application before your embedded target hardware is available using the optional esd EtherCAT interface ECS-PCIe/1100 which comes with support for the identical (cross-platform) API for Windows®/Linux®.

Additional Services (not scope of delivery)

- General EtherCAT technology seminars. If requested, the seminar can also be held on-site.
- Workshop for detailed introduction to the EtherCAT slave stack code with the possibility to adapt the code to the target device.
- Technical consulting before and during the development phase of the EtherCAT slave device. Our customers will benefit from our years of experience in EtherCAT hardware and software development.

Technical Specifications:

General:	
Header files:	ANSI C
Build:	Includes sample makefiles for gcc and project files for Visual Studio 10

Source Code Version:	
Language:	Completely written in ANSI C
Other:	Little/big endian compatible
Resource usage*:	RAM: approx. 5 kB (BSS/DATA/Heap) + 2 kB (Stack) ROM: approx. 15 kB (CODE/CONST)

*EtherCAT slave application with 64 byte process data, CoE with 'SDO Information Service' support and approx. 30 objects total. (No EoE etc.)

Order Information:

Designation		Order no.
EtherCAT Slave Stack Source	EtherCAT Slave Stack ANSI-C source code version	P.4520.01
Optional Hardware for Rapid Prototyping		
ECS-PCIe/1100	PCI Express board with EtherCAT slave controller ET1100, incl. driver, stack binary and documentation for Windows and Linux on CD	E.1100.02