



Ether CAT

Connecting dSPACE systems to
EtherCAT networks

EtherCAT is an Ethernet-based field bus with many uses. One of them is as a communication system for applications in industrial automation. The new EtherCAT Slave Interface integrates dSPACE systems into EtherCAT networks.

Application Area

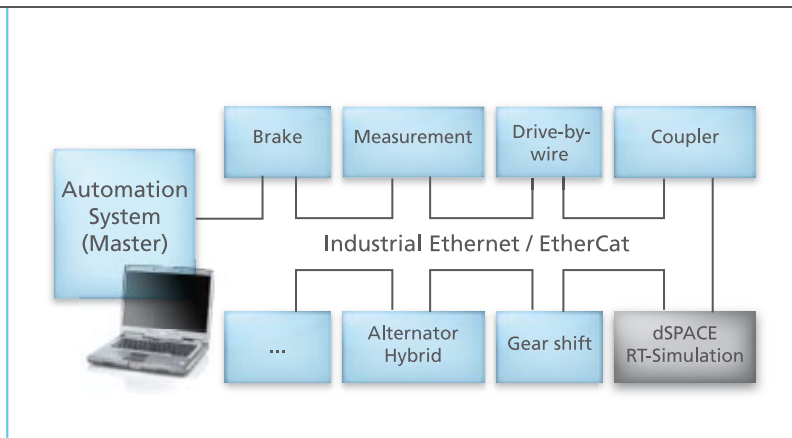
EtherCAT is an open protocol that is particularly suitable for implementing real-time applications in industrial automation. The protocol supports numerous network topologies such as line, tree, ring, star and combinations of these. The new EtherCAT Slave Interface has been specially designed to integrate a dSPACE system into an EtherCAT network as a slave. A typical application is an engine test bench with an integrated dSPACE System.

Hardware Concept

The EtherCAT Slave Interface is based on the DS5202 FPGA Base Board, the EV1039 piggy-back module, and either one or two Beckhoff FB1111-0140 EtherCAT controller boards. The DS5202 acts as the interface between the PHS bus and the dual-port memory (DPMEM) on the modules. The DPMEM is also addressed by the bus. To ensure data consistency, the EtherCAT Slave Controller performs synchronization.

Software Concept

The EtherCAT Slave Interface is configured by means of an M script that defines the object directory of the



Example of an engine test bench whose components are connected via an EtherCAT network.

Glossary

Ethernet –
Fast, cable-bound data network

Field bus –
Industrial communication system for connecting a large number of devices, actuators, and sensors

Mailbox –
Memory for CANopen messages

slave and some global settings. A Simulink template model and the device description file (DDF) are then created from these with the aid of an associated generator. The blocks in the template model are completely preconfigured and can be dragged straight to the function model for insertion. This method ensures consistency between the application on the dSPACE system and the DDF that is made available to the EtherCAT master.

Interaction and Communication

The EtherCAT Slave Interface supports the following methods for interaction and communication, and also pro-

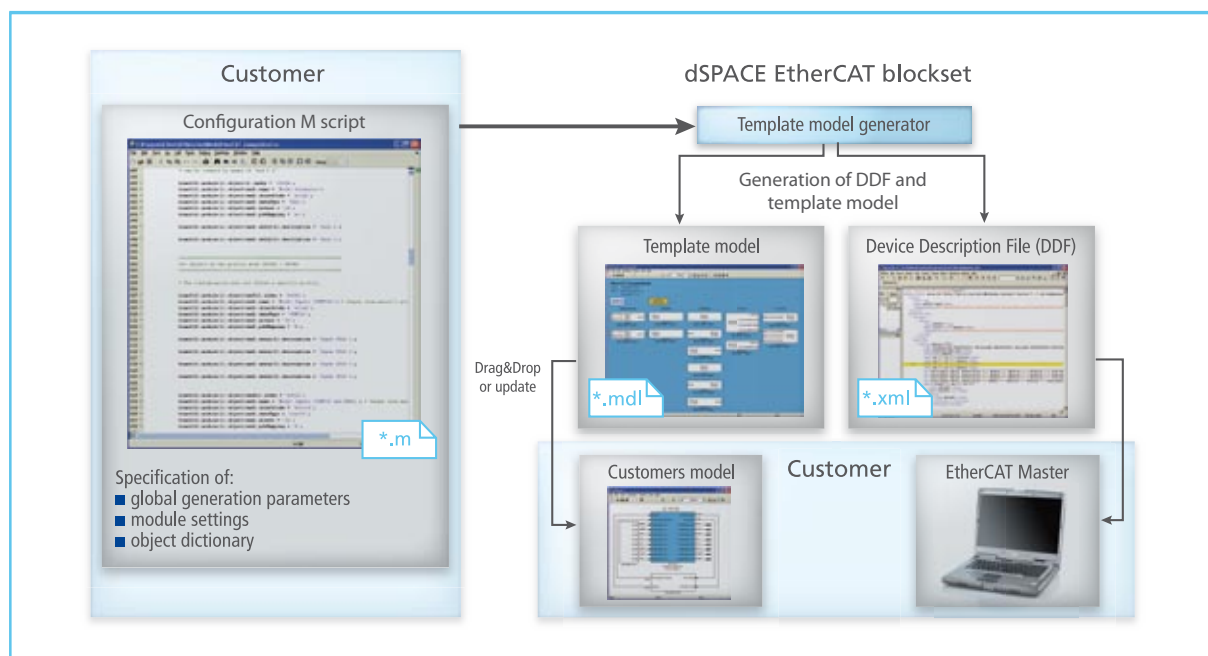
vides the necessary Simulink blocks:

- State machine
- Interrupts
- CANopen over EtherCAT (CoE)
- Cyclic communication (RX, TX)

The state machine block processes the master's request to transition to a new state. It checks the validity of the requested transition and performs the associated actions.

The interrupt block passes EtherCAT interrupts to the function model so that actions can be activated synchronously. Interrupts can be selected and masked without the template model having to be regenerated. The solution also supports acyclic

communication with CANopen over EtherCAT. The object directory can be read from within the master, and individual entries can also be written. CANopen transfers can be executed in a background task. The mailbox provides 512 bytes for each direction and supports segmented transfers. A callback mechanism can be used to serve master accesses to the object directory with user C code. Six sync managers are available for cyclic communication. This means that different cycle times can be implemented to optimize the bus load. The configuration of the process data communication is static during run time. ■



Starting with an M script, the communication blocks and EtherCAT device description file are created.