CAN/CANopen to EtherCAT Gateways: Requirements and Solutions

Martin Rostan, Beckhoff Automation GmbH
Mirko Tischer, Vector Informatik GmbH

CAN is the dominating network technology in automotive applications. Enhanced by the higher layer protocol CANopen it is a well established Fieldbus technology, implemented in a large variety of devices and systems. EtherCAT is an Industrial Ethernet technology which provides high end communication performance, flexible topology options and low system costs. In many applications, the distinct advantages of both networks have to be combined. With CAN/CANopen to EtherCAT gateways, this can be done efficiently, if certain design rules are followed.

This paper discusses the requirements on such gateways from the application and from the device vendor point of view. Besides CANopen gateways, the special requirements of generic CAN to EtherCAT gateways as used e.g. in automotive test bed applications are considered. It is shown that CANopen protocols can be used to configure the gateway also from the EtherCAT side. Example implementations and their representation in software tools are shown as well as application examples. It is also shown that such gateways enable a smooth migration path from CANopen devices towards Industrial Ethernet.

EtherCAT Overview

EtherCAT was originally developed by Beckhoff and introduced in 2003. Meanwhile the technology is an IEC standard [2-4], and supported by the EtherCAT Technology Group [7], an international user and vendor organization with over 650 member companies from 37 countries (as of Dec 2007). EtherCAT utilizes CANopen [6] communication protocols such as SDO and PDO and also supports CANopen device profiles. In IEC 61800-7-301 [5] the mapping of the CANopen servo drive profile CiA402 onto EtherCAT is standardized. EtherCAT employs the functional principle of “processing on the fly”: unlike other Industrial Ethernet technologies, the Ethernet frame or packet is no longer received, then interpreted and process data then copied at every device. The EtherCAT slave devices read the data addressed to them while the frame passes through the node. Similarly, input data is inserted while the telegram passes through. This leads to superior bandwidth utilization and thus to short cycle times – EtherCAT is considered to be the fastest Industrial Ethernet technology – while allowing for flexible topology options and low implementation and system costs. Typical EtherCAT communication cycle times are 0.1 – 1ms, the network extension, number of nodes and the topology options are almost unlimited. A more detailed introduction into EtherCAT and its usage of CANopen protocols can be found in [1].

EtherCAT masters do neither require any special hardware nor a dedicated communication processor and can be implemented in software on any control hardware that provides an Ethernet port. EtherCAT Slave Controller chips are available from several manufacturers.

Gateways

CAN / Ethernet gateways have been discussed for quite some time (e.g. in [8, 9, 10, 11]). The system architecture behind these approaches is hierarchical (figure 1): field devices networked locally with CAN or CANopen fieldbus systems are connected with the Ethernet based control or management level via gateways. This initial approach avoids complex timing requirements both for Ethernet and the gateway, since all real time control loops are closed locally within the CAN environment. The Ethernet connection is used for non real time tasks such as data acquisition, remote configuration and diagnosis.

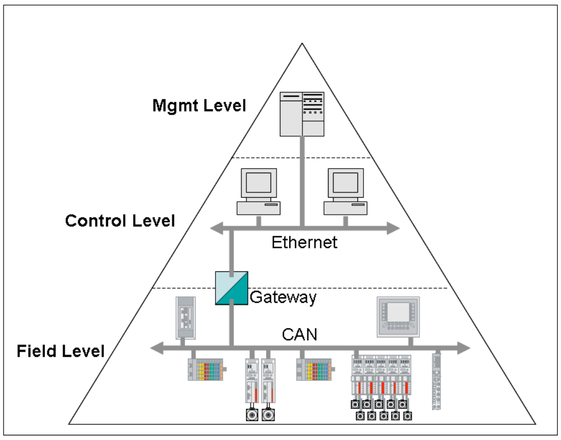


Figure 1: Hierarchical Control Architecture

The system architecture and thus the requirements for gateways change with the arrival of real time Ethernet technologies such as EtherCAT (see figure 2). Now the Ethernet based system does not only reach the real time characteristics (such as reaction and cycle time) of the CAN based network, but exceeds these performance parameters substantially. At the same time EtherCAT overcomes one of the main constraints of CAN: limited network extension particular at high baud rates.

And since EtherCAT interfaces can also be implemented at low costs, the application range of the Ethernet technology is enhanced towards and beyond the fieldbus level, reaching the embedded system level.

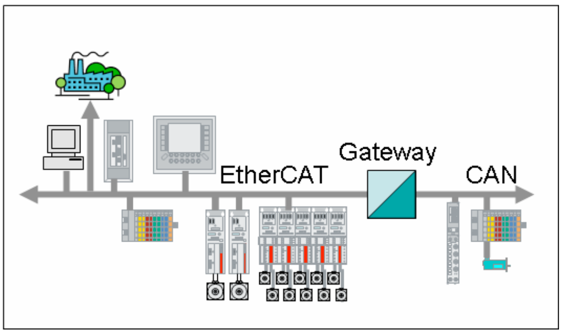


Figure 2: Flat Control Architecture

The real time domain is no longer concluded in the CAN network, but spans across the network technologies. The requirements on the gateways change accordingly, since CAN networks are now used as local extensions of the real time EtherCAT system.

CANopen to EtherCAT gateway

CANopen networks are used in a very broad range of application fields such as machine control, medical devices, off-road and rail vehicles, maritime electronics, building automation as well as power generation. When considering CANopen to EtherCAT gateways, the following CANopen device classification may help to sort the applications:

Fieldbus devices such as I/O-blocks, drives, sensors, actuators, valves etc. which typically are also available with other common fieldbus interfaces. Here the CANopen to EtherCAT gateway predominantly serves the purpose of integrating such CANopen devices which are not yet available with EtherCAT interface.

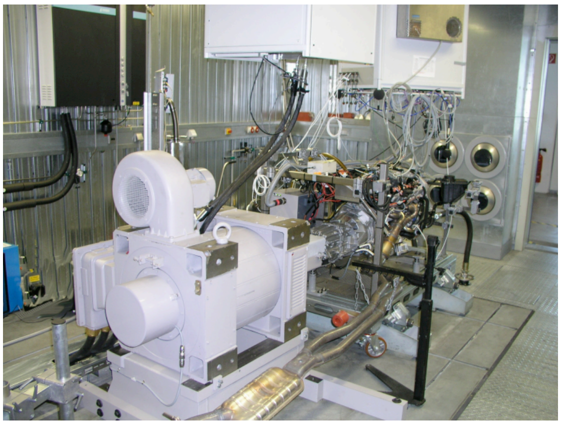
Embedded devices such as small embedded controllers, custom made sensor interfaces, specialized hardware components for machine control used in conjunction with general purpose fieldbus devices. For special purpose control subsystems – often they include the machine builders dedicated process know how – CANopen so far has been the network of choice, since the CAN hardware is simple to integrate and CANopen provides the interoperability. While EtherCAT has similar features and will thus further expand in this environment, gateways here are very useful since one tries to avoid the re-design of such special hardware if possible even if CANopen is replaced by EtherCAT as main control network.

Deeply embedded devices in vehicles, medical devices etc, where no standard “off the shelf” devices can be used. In conjunction with such systems gateways are typically used to enable the classical hierarchical architecture: local, widely independent networks have to be connected to supervisory or management levels equipped with EtherCAT. However, EtherCAT is already used as backplane bus systems and hence in deeply embedded applications. And since EtherCAT slave controllers are also available with extended temperature ranges, this network is also beginning to even enter deeply embedded systems with rigid environmental requirements such as mobile machines.

The gateway timing requirements in fieldbus and embedded device scenarios typically are more demanding than in applications where deeply embedded systems have to be connected. Whenever the long term goal is to move from CANopen to EtherCAT, gateways provide a smooth migration path.

CAN to EtherCAT gateway

Most in-vehicle applications do not use CANopen, but proprietary protocols on top of the CAN physical and data link layer. Generic, non-standardized CAN protocols are also used in many deeply embedded applications. Unlike with CANopen, higher layer protocol information may also be encoded in the data length code or identifier field of the CAN frame, so forwarding the CAN payload data through the gateway may not be sufficient. In addition, the CAN network timing itself may contain valuable information (such as occurrence of certain signals), so that additional timing information may have to be provided, too.



Picture 1: Engine Test Bed (Photo: BMW AG)

CAN to EtherCAT gateways are used in automotive test bed applications, e.g. at the BMW Research and Innovation Center in Munich, Germany (see picture 1+2). EtherCAT was chosen as automation network for the new engine test center that starts operations later this year, and some testing equipment is connected via Gateways. The ability to integrate remote CAN interfaces without jeopardizing the performance on either side was a crucial criterion for the network selection in this application. Connecting in-vehicle CAN networks used for engine control via Gateways is also possible.

Picture 2: BMW Research + Innovation Center (FIZ) in Munich, Germany (Photo: BMW AG).

Gateway design considerations

When designing such a gateway the specific requirements of coupling EtherCAT and CAN have to be taken into account, in order to maximize performance, achieve the best possible synchronization and thus prepare the gateway for the widest range of applications.

Data throughput

The data throughput of EtherCAT and CAN is different by orders of magnitude. While CAN transports up to 8 bytes per frame at a maximum of 1 MBit/s (60 Kbytes/s), EtherCAT transmits up to 1486 bytes per frame at 100 MBit/s (10.000 Kbytes/s with the possibility of a future extension to Gbit Ethernet). EtherCAT can transport the bus traffic of several CAN networks without generating an overload situation, if several CAN frames per EtherCAT frame are transmitted.

Cyclic vs. Event driven communication

Another big difference between the systems is how the transmission of the messages is triggered. With EtherCAT the frame is always sent by the master. The EtherCAT master typically operates cyclically and therefore the communication typically is directly linked with this cycle. On the CAN side communication is often event driven. Whenever data has to be transmitted, the device initiates the sending of a frame. The differences in data throughput and in message triggering between the systems in any case require buffer mechanisms in the gateway.

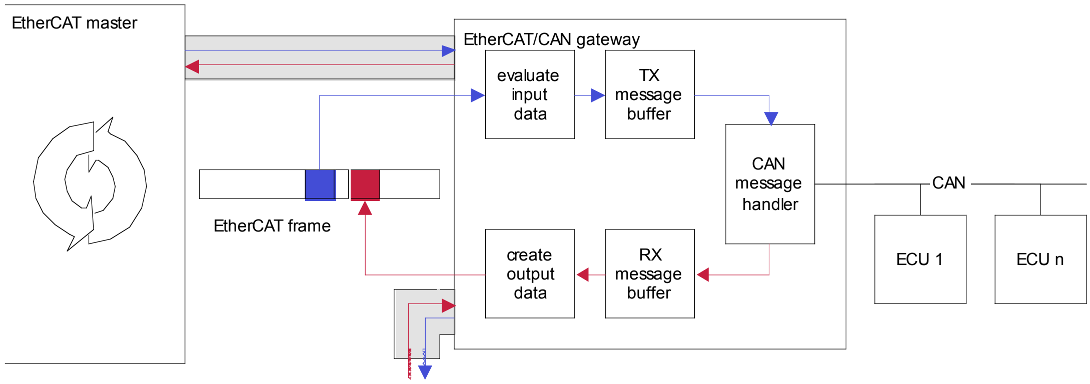


Figure 3: Gateway System Structure

The throughput of a gateway concept based on figure 3 is determined by the following parameters:

- Size of the message buffer in the gateway
- Amount of CAN-Data transmitted within one EtherCAT frame
- Cycle time of the EtherCAT Communication

The gateway has to be designed, integrated and configured in such a way that no CAN frames are lost. For the reception of the CAN frames this means that the message buffer inside the gateway has to be large enough to store all CAN frames received within one EtherCAT cycle.

For timing considerations we assume the following boundary conditions. The CAN bus is operated at 1 Mbit/s. The transmission of a CAN message with one byte takes about 55µs. If one assumes an EtherCAT cycle time of 1000µs, in worst case 18 CAN frames can be received within this period of time. So the receive buffer at least has to be able to handle this number of CAN frames.

Furthermore, within one EtherCAT Frame the content of at least of 18 CAN frames has to be transmitted to the EtherCAT master, since otherwise a buffer overflow may occur if the CAN busload is close to 100% over several cycles. Accordingly, Table 1 shows the buffer size requirements for CAN with 1 Mbit/s and messages with one data byte.

Table 1: Relation between EtherCAT-Cycle and buffer size for CAN Frame Reception

EtherCAT Cycle (µs)	Minimal required Buffer Size (CAN-messages)
1000	18
2000	36
4000	72

While transmitting the data of the received CAN frames over EtherCAT the reception time of the CAN frame is lost. If this timing information is valuable – in the general case it is – the Gateway has to provide a reception time stamp for each received CAN frame.

The send throughput of course also depends of the chosen buffer size. The CAN data first has to be extracted from the EtherCAT frame. The associated CAN frame can only be sent once the CAN bus is available (bus idle). Therefore a send buffer is also required. For emptying this buffer (the CAN send process) the gateway has to provide means to avoid message bursts and thus high bus loads. This is in particular important at higher CAN bit rates, since the message burst may exceed the frame handling capabilities (interrupt load) of the connected CAN devices.

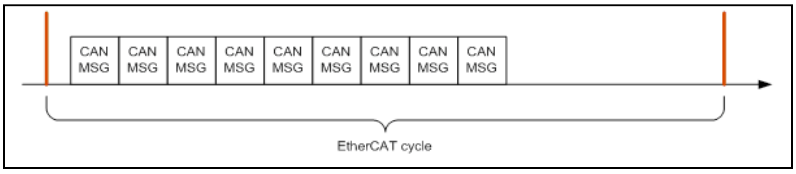


Figure 4: cyclic CAN frame burst

A burst as shown in figure 4 would occur in each EtherCAT cycle if the EtherCAT frame transports the data of more than one CAN frame. Since with longer EtherCAT cycles this behavior is more likely to become problematic, the gateway should be able to insert gaps in between the CAN frames that it sends.

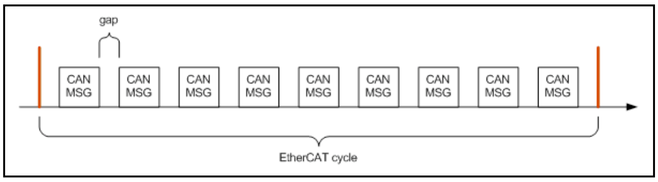


Figure 5: Gaps inserted in the send frame sequence

In case such gaps are inserted in between the Send- Frames it may happen that not all CAN frames can be sent within one EtherCAT cycle. This may also occur if the CAN bus is busy with higher priority frames so that the gateways does not get sufficient bus access with CAN frames of lower priority. Therefore in send direction there has to be a flow control mechanism in order to ensure that there is no buffer overflow with associated send frame loss.

Process Data-Gateway vs. Message-Gateway

When implementing such a gateway there are two main approaches: a process data-gateway only transmits the payload (process data or signals) of the CAN frame via EtherCAT.

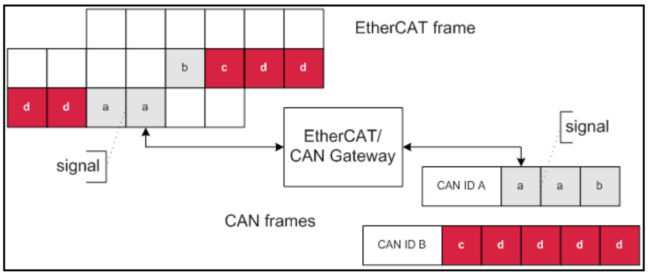


Figure 6: Signal Gateway Approach

The advantage of this approach is, that the EtherCAT master can directly provide the signals in form of a process image (see figure 6). The disadvantage is that the process data description has to be configured in the gateway itself. In case of CANopen, where the process data allocation and configuration is part of the standard (PDO mapping) this can be implemented inside the gateway. The gateway then translates the signals into CAN frames respective interprets received CAN frames based on this configuration. Furthermore, for data consistency reasons all data received within one CAN frame has to be transported within one EtherCAT frame, which can be difficult to ensure with this methodology if the CAN higher layer protocol is unknown. So while this approach is simple from the application point of view, it only works if the CAN protocol stack is well defined and implemented inside the Gateway.

Therefore the message based approach is preferable for generic CAN / EtherCAT gateways while the process data or signal oriented approach suits the requirements in CANopen to EtherCAT gateways better.

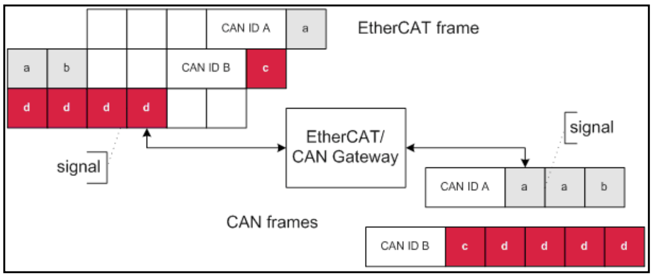


Figure 7: Message Gateway Approach

Other than with the process data-gateway entire CAN frames are transported within one EtherCAT frame (see figure 7). Inside the EtherCAT frame there is a data container in which the CAN frames and the associated management data (such as number of frames within the container, status info, flow control data etc.) is located.

Since each EtherCAT frame may contain different CAN messages, the data cannot be made available directly as process image. Instead the master has to interpret the data first in order to copy it into the process image. While the gateway becomes much simpler, the effort is moved into the EtherCAT master or in the application layer of the master.

The message gateway approach is much more flexible: without modifying the gateway, any CAN based protocol can be used.

Configuration: Modular Device Profile

The gateway either provides the CAN frame queues towards EtherCAT (message gateway, object dictionary layout see table 2) or represents a number of CANopen devices (process data gateway, see table 3). It is typically configured from the EtherCAT side using the CoE (CANopen over EtherCAT) protocol with the modular device profile approach. This versatile profiling model was developed by the EtherCAT Technology Group. It allows one to configure devices which have a dynamic parameter set (object dictionary) by writing the expected module configuration at boot-up. Alternatively it can be generated automatically after power-on.

Furthermore, the modular device profile gives transparent process data and parameter access to underlying modules such as the devices connected via gateway. Details regarding the modular device profile can be found in [12].

Table 2: Object Dictionary Layout for CAN to EtherCAT Gateway (message gateway)

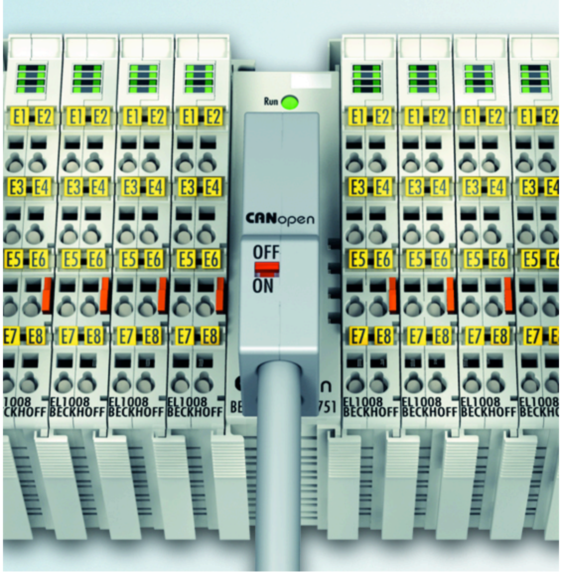
Index (hex)	Object Dictionary Area
0x0000-x0FFF	Data Type Area
0x1000-x1FFF	Communication Area
0x6000-x6FFF	Input Area (CAN RX message queue)
0x7000-x7FFF	Output Area (CAN TX message queue)
0x8000-x8FFF	Configuration Area (configuration of the CAN Interface)
0xF000-xFFFF	Device Area

Table 3: Object Dictionary Layout for CANopen to EtherCAT Gateway (process data gateway)

Index (hex)	Object Dictionary Area
0x0000-x0FFF	Data Type Area
0x1000-x1FFF	Communication Area
0x6000-x6FFF	Input Area (TxPDOs of the CANopen slaves)
0x7000-x7FFF	Output Area (RxPDOs of the CANopen slaves)
0x8000-x8FFF	Configuration Area (Expected configuration of the CANopen slaves)
0x9000-x9FFF	Information Area (Detected configuration of the CANopen slaves)
0xF000-xFFFF	Device Area

Implementation Example

Both gateway approaches (CAN and CANopen to EtherCAT) have been implemented e.g. in the Beckhoff EtherCAT Gateway terminal EL 6751 (see picture 3). Its CANopen functionality corresponds to the Beckhoff PCI master card FC5101. It is either available as CANopen master or slave device. The master version also supports generic CAN together with message timestamping, flexible bit timing, extended message queues and filter functionality.



Picture 3: CAN/CANopen to EtherCAT Gateway Terminal with I/O EtherCAT Terminals.

Performance Considerations

A CAN/CANopen to EtherCAT gateway can be considered to be a PCI CAN/CANopen card which was removed from the PCI bus and placed remotely outside of the PC chassis. Due to EtherCATs performance the update rates and bandwidth of both variants is comparable, typically the EtherCAT update rate is even better. Therefore the gateway connection does not provide a bottleneck any more, EtherCAT allows one to connect to a CAN/CANopen network without performance restictions.

Summary

In a growing number of applications CAN and CANopen networks have to be interfaced to Industrial Ethernet networks such as EtherCAT. The required gateways face challenges regarding transparency, performance, simple configuration, costs and of course standardized uniform software interfaces. It makes sense to distinguish between the CAN and CANopen gateway requirements, since they lead to different architectural approaches. A generic CAN gateway, for which the CAN higher layer protocol is a priori unknown, a message based approach fits best. For a CANopen gateway, one can implement the protocol stack in the gateway and provide a process image interface on EtherCAT.

Equipped with EtherCAT such gateways are no bottleneck but provide a full performance connection to CAN or CANopen networks. Hence they also build a smooth migration path from CANopen to EtherCAT.

References

[1] M. Rostan: "CANopen over EtherCAT – taking a CAN technology to the next level", Proceedings of the 11th International CAN Conference, pp. 12/1-12/8, 8-9 March 2005, Rome, Italy

[2] IEC 61158-2 (Ed.4.0), Industrial communication networks - Fieldbus specifications - Part 2: Physical layer specification and service definition

[3] IEC 61158-3/4/5/6-12 (Ed.1.0), Industrial communication networks - Fieldbus specifications - Part 3-12: Data-link layer service definition - Part 4-12: Data-link layer protocol specification - Part 5-12: Application layer service definition - Part 6-12: Application layer protocol specification -Type 12 elements (EtherCAT)

[4] IEC 61784-2 (Ed.1.0), Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3 I

[5] IEC 61800-7-301/304 (Ed.1.0), Adjustable speed electrical power drive systems - Part 7-301: Generic interface and use of profiles for power drive systems - Mapping of profile type 1 to network technologies - Part 7-304: Generic interface and use of profiles for power drive systems - Mapping of profile type 4 to network technologies

[6] EN 50325-4: Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces. Part 4: CANopen.

[7] EtherCAT Technology Group website, http://www.ethercat.org

[8] H. Ekiz, A. Kutlu, M.D. Baba, E.T. Powner: "Design and Implementation of a CAN/Ethernet Bridge", Proceedings of the 3rd International CAN Conference, pp. 11/17-11/26, 1-2 October 1996, Paris, France

[9] H. Ekiz, A. Kutlu, M.D. Baba, E.T. Powner, "Performance Analysis of a CAN / Ethernet Bridge", IEEE SICON, 97 Conference, pp. 71-85, 14-17 April 1997, Singapore

[10] CiA309 draft standard series: "Interfacing CANopen with TCP/IP", CAN in Automation e.V.

[11] H. Zeltwanger: "Embedded and deeply embedded networks for machine internal controls", Proceedings of the SPS/IPC/Drives Congress, pp. 443-449, 2003, Nuremberg.

[12] ETG.5001: "Modular Device Profile Specification", EtherCAT Technology Group

Martin Rostan
Beckhoff Automation GmbH
Ostendstr. 196, D- 90482 Nürnberg
Phone +49 (911) 54056-11
Fax +49 (911) 54056-29
m.rostan@beckhoff.com
www.beckhoff.com

Mirko Tischer
Vector Informatik GmbH
Ingersheimer Str. 24, D-70499 Stuttgart
Phone +49 (711) 80670-0
Fax +49 (711) 80670-111
mirko.tischer@vector-informatik.de
www.vector-informatik.com