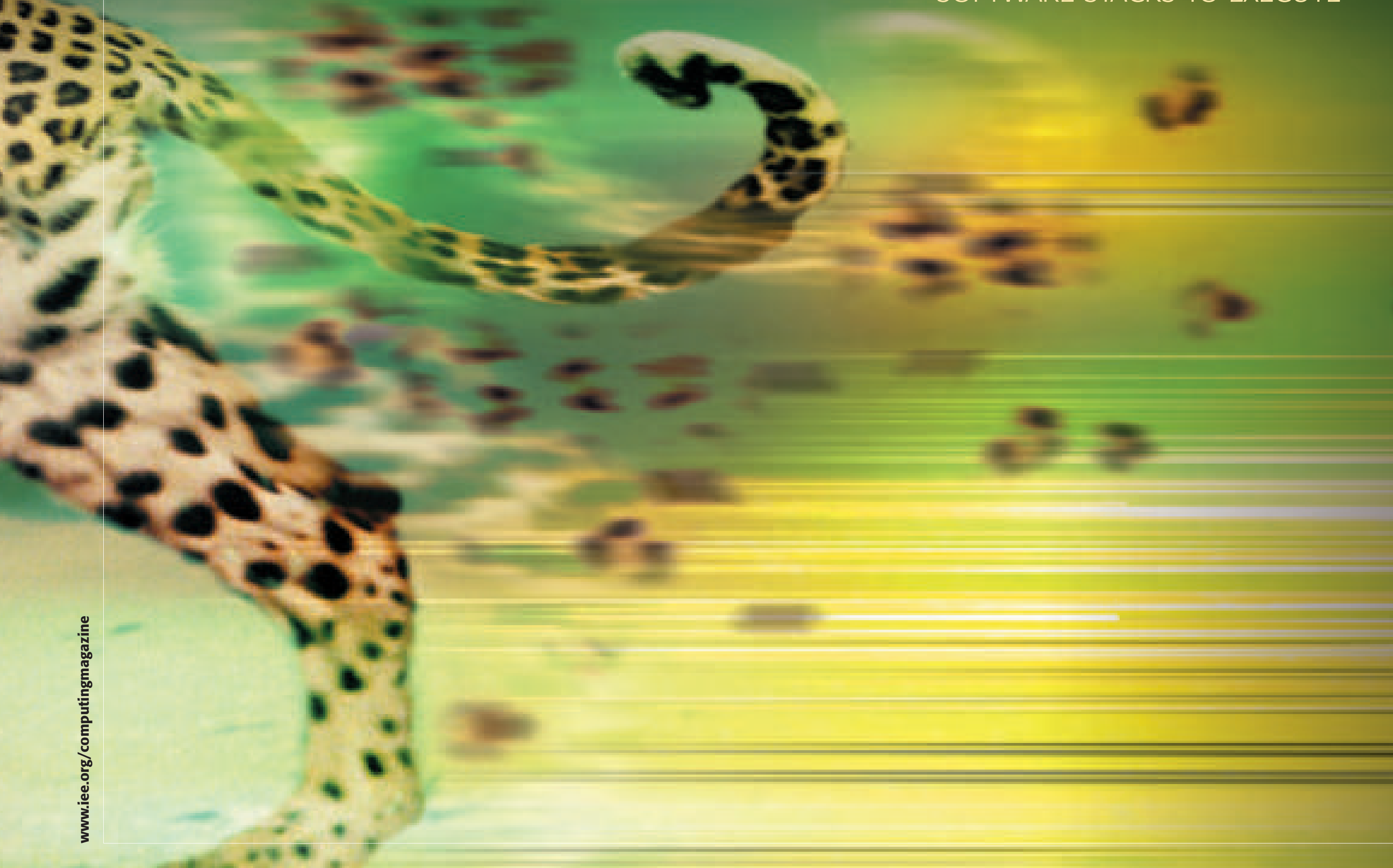
Real-Time Ethernet the EtherCAT solution

By Dirk Jansen and Holger Büttner

ETHERCAT ACHIEVES ITS HIGH SPEED BY PERFORMING ALL COMMUNICATION FUNCTIONS IN HARDWARE – THERE ARE NO MICROPROCESSORS OR SOFTWARE STACKS TO EXECUTE



The use of Ethernet as the physical communication layer for automation technology is increasing worldwide. Ethernet is already established at the command level, for factory networking, and for inter-control communication.

At the field level, fieldbuses developed in the 1990s continue to dominate the market. With maximum gross transfer rates of 0·5-16MB they continue to be adequate for many applications, however their costs remain high, particularly for the wiring. But the fieldbus organisations and large automation companies are moving on: the majority of them have chosen Ethernet as the future extension or replacement for their proprietary fieldbus technologies.

It must be admitted, Ethernet at the field level has its problems. It is not optimised to send frequent, short messages. The connection costs are high, and require microprocessors at each node. Running software in these microprocessors slows the system down, so it can't achieve real-time capability. The commonly used star topology for switched Ethernet leads to excessive cabling and highly cascaded communication dependencies.

To address some of these issues, Beckhoff Industrie Elektronik has developed a new approach to bringing Ethernet to the I/O domain. The new system is called EtherCAT (Ethernet for Control Automation Technology), and the system is very fast: it can process 12,000 digital I/O in 350 microseconds, for instance, or 100 servo axes in 100 microseconds.

The development of EtherCAT mainly focused on the following targets:

- *Broad applicability.* Any controller with an Ethernet port can be an EtherCAT master. Any computer can be an EtherCAT control system without special connections.
- *Full conformity with the Ethernet standard.* EtherCAT coexists with other Ethernet devices and protocols on the same bus. Standard structural components such as Ethernet switches are usable with EtherCAT.
- *Smallest possible device granularity without having to use a sub-bus.* Any type of I/O node, from complex nodes to 2-bit I/Os, can be used economically as EtherCAT slaves.
- *Maximum efficiency.* As much of the Ethernet bandwidth as possible should be available for user data transfers.
- *Short cycle times.* Cycle times significantly less than 100 microseconds open up new areas of application, such as closing servo control loops.
- *Maximum deterministic properties*. Even without the need for absolute telegram transfer accuracy.

ETHERCAT OPERATING PRINCIPLE

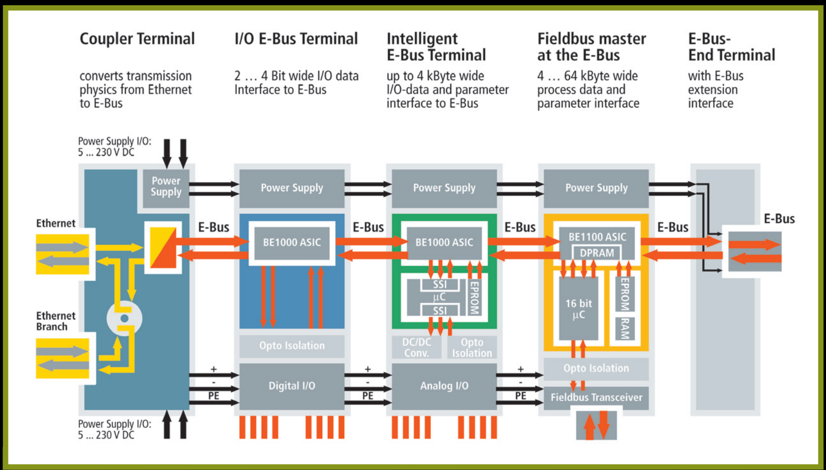
From an Ethernet point of view, an EtherCAT bus is simply a single large Ethernet device that sends and receives Ethernet telegrams. However, the 'device' does not contain an Ethernet controller with downstream microprocessor, but a large number of EtherCAT slaves. These process the incoming telegrams directly and extract the relevant user data, or they insert data to the telegram and transfer it to the next EtherCAT slave. The last EtherCAT slave sends the fully processed telegram back to the master, as a response telegram.

This procedure utilises the fact that Ethernet deals separately with transfers in separate directions (Tx and Rx lines) and operates in full duplex mode: the transmitted telegrams are returned to the control by loop-back through the Rx wire pair. Like any other Ethernet device, direct communication without a switch may be established using a 'crossover' Ethernet cable, thereby creating a pure EtherCAT system.

TELEGRAM PROCESSING

Telegrams are processed directly 'on the fly' at each I/O node. While the telegrams (delayed by only a few bits) are passed on, the slave recognises relevant commands and executes them accordingly. Processing is done within the hardware and is therefore independent of the response times of any microprocessors that may be connected. Each device has an addressable memory space of 64kByte that can be read or written, either consecutively or simultaneously.

Several EtherCAT commands can be embedded →



This diagram shows the Ethernet signal arriving from the left into the coupler terminal, where it is converted into an LVDS signal for the E-Bus, which is the backplane bus for the terminals. The first and second terminals are digital and analogue. In each terminal the signal goes into the FMMU chip, which is marked 'BE 1000 ASIC' on this diagram. The third terminal is a fieldbus master (it could be Profibus or CANbus, for example) and the FMMU in this case couples via a dual ported RAM to a microcontroller that accomplishes the protocol transfer to the specific fieldbus. At the end is an E-bus end terminal with an RJ-45 connector that can lead to another terminal or the signal can be reflected back. Notice the signal can also be continued to another set of terminal blocks through the lower Ethernet connection in the coupler terminal. An intelligent router switch in the coupler terminal (the white round circle) knows whether to send the signal back to the PC through the upper Ethernet connection or to route it to an Ethernet branch.

within an Ethernet telegram, each addressing individual devices and/or memory areas. EtherCAT commands are transported in the data area of the telegram and can either be coded via a special Ether type or optionally via UDP/IP.

The first variant uses a special Ether-type. Its use is limited to an Ethernet subnet, that is, associated telegrams are not relayed by routers. For control tasks this usually does not represent a constraint, and, the installation meets lowest cost possible, is simple, and it avoids all issues with MAC-IDs and IP addresses in the field (while maintaining full IEEE 802.3 compliance).

Each EtherCAT command consists of an EtherCAT header, the data area and a subsequent counter area (working counter), which is incremented by all EtherCAT devices that are addressed by the EtherCAT command and have exchanged associated data.

The second variant uses UDP/IP. It generates a slightly larger overhead (IP and UDP header), but for less time-critical applications it enables normal IP routing to be used. It requires the first EtherCAT device to provide a MAC-ID for routers and switches.

On the master side, TCP/IP stacks can also be used.

FIELDBUS MEMORY MANAGEMENT UNIT (FMMU)

Using the telegram structure described above, several EtherCAT devices can be addressed via a single Ethernet telegram with several EtherCAT commands, which leads to a significant improvement of the usable data rate. However, for 2 bit input terminals with precisely 2 bits of user data, the overhead of a single EtherCAT command is still excessive.

The Fieldbus Memory Management Unit eliminates this problem and the available data rate to be utilised is almost 100% – even for devices with only 2 bits of user data. Similar to the Memory Management Units (MMU) in modern processors, the FMMU converts a logical address into a physical one via an internal table. The FMMU is integrated in the EtherCAT slave ASIC (Application-Specific Integrated Circuit) and enables individual address mapping for each device.

The FMMU also supports bit-wise mapping. This enables the two bits of the input terminal to be inserted individually anywhere within a logical address space. If an EtherCAT command is sent that reads or writes to a certain memory area, instead of addressing a particular EtherCAT device, the 2 bit input terminal inserts its data at the right place within the data area. All other terminals that detect an address match within their own FMMU insert their data, so that many devices can be addressed simultaneously with a single EtherCAT command.

Since the FMMU is present in each device and is configured individually, the master can assemble complete process images during the initialisation phase and subsequently exchange them via a single EtherCAT command. Additional mapping is no longer required in the master, so that the process data can be assigned directly to the different control tasks (PLC, NC, etc.) Each task can create its own process image and exchange it within its own time frame. The physical order of the EtherCAT devices is completely independent and is only relevant during the initialisation phase.

ETHERCAT IS INDEPENDENT OF THE TRANSMISSION SPEED – IT CAN BE USED FOR 1GB OR MORE WITHOUT MODIFICATION

The logical address space for the FMMUs is 4GB. EtherCAT is therefore a type of serial backplane for automation systems that enables connection to distributed process data for both large or very small automation devices. Via a standard Ethernet controller and a standard Ethernet cable (CAT 5), practically any number of I/O channels can be connected to automation devices, which can be accessed with high bandwidth, minimum delay and near-optimum usable data rate.

ETHERCAT IS SO FAST THAT USUALLY THE RESPONSE RETURNS TO THE MASTER WHILE THE MASTER IS STILL SENDING THE LAST BYTES OF ITS QUERY

PHYSICAL LAYER

A wide range of physical Ethernet media is available, starting from the good old 'yellow cable' to high-speed fibre-optic lines. In addition to the classic CAT-5 copper cables (100 base TX), the 100MB Ethernet used by EtherCAT also offers fibre-optic physical layers (100 base FX). Since EtherCAT is fully compatible with Ethernet, all associated physical layers can be used.

EtherCAT transmissions often involve very short distances, for example between two EtherCAT terminals within the same terminal block, so, for economy, an additional physical layer was developed – the E-bus. The E-bus is based on LVDS transmission (Low Voltage Differential Signal, IEEE standard P1596.3-1995). Apart from terminal communication, it is also suitable for cost-effective "long-distance transmissions" up to approximately 10m.

Since all transferred data consist of fully compatible Ethernet telegrams, the physical layer can be changed anywhere and any number of times. In a system consisting of different control cabinets and machine modules, for example, for each unit the most cost-effective physical layer can be used. Within a control cabinet the E-bus is sufficient; between the cabinet and the machine modules standard Ethernet copper technology is used up to distances of 100m. For even larger distances or extreme EMC loads, fibre-optic technology can be implemented anywhere within the system.

The only prerequisite for the transmission medium is full duplex capability. This is because EtherCAT is so fast that usually the response is sent back to the master while the master is still sending the last bytes of its query. Half duplex physical layers as used in radio transmission, for example, would cause collisions.

In the current implementation, EtherCAT utilises 100MB Ethernet. However, the transmission principle is independent of the speed and can be used in the future for 1GB or more without modification.

TOPOLOGY

The topology of a communication system is one of the crucial factors for the successful application in the automation industry. The topology has significant influence on the cabling effort, diagnostic features, redundancy options and hot-plug and-play features. While the star-shaped cabling commonly used for standard Ethernet has advantages with regard to hot-plug-and-play and redundancy, the cabling effort and switches required in distributed applications with many devices are not really acceptable.

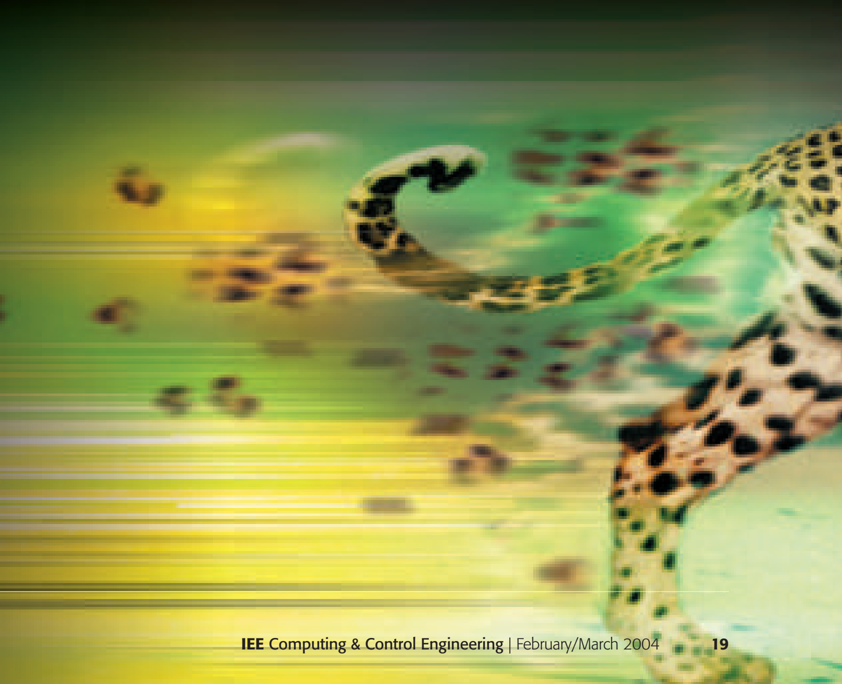
In logical terms, in EtherCAT the slaves represent an open ring bus. At the open end, the master sends in telegrams, either directly or via standard Ethernet switches, and receives them at the other end after they have been processed. All telegrams are relayed from the first device to the next. The last device returns the telegram back to the master. Since a normal Ethernet cable is bidirectional (separate Tx and Rx cables), and since all EtherCAT slaves can also transfer in the reverse direction, the result is a physical line.

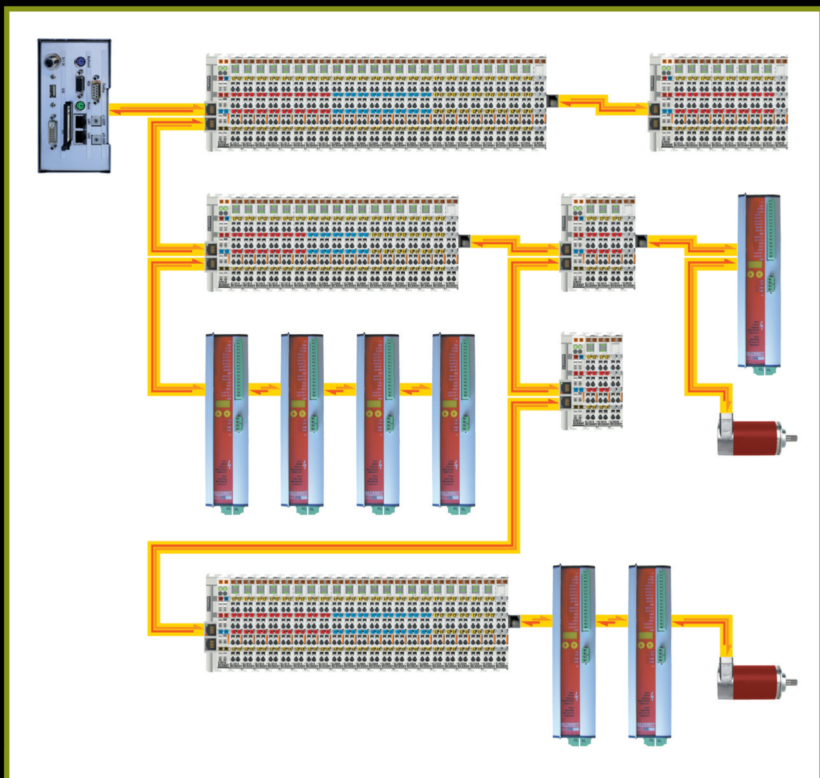
Branches, which in principle are possible anywhere, can be used to build a flexible tree structure from the line structure. A tree structure enables very simple wiring; individual branches, for example, can branch into control cabinets or machine modules, while the main line runs from one module to the next.

DIAGNOSTICS

For the operation of a distributed bus system, diagnostic options are no doubt just as significant as performance data, topology features or cabling effort. In this respect EtherCAT meets the expectations of a modern communication system.

Unlike 'party line' bus systems (e.g. Profibus or CAN), in which all devices are connected to the same physical cable, and the signals are sent to all devices without being refreshed, 100MB Ethernet and EtherCAT use pure point-to-point transfers. Faults or even sporadic weak points that are impossible to trace in party line systems, or only ➔





This EtherCAT system shows the Ethernet signal originating from the industrial controller at the upper left. It is routed through a number of terminal blocks, servo drives and servo motors. The first coupler terminal converts Ethernet to an E-bus signal for all subsequent terminals. The signal can be converted back to Ethernet at any point, but this only needs to be done if the terminals are more than 10m apart. Two Ethernet connections on the left side of each coupler give maximum flexibility for arranging the terminals.

with special measurement arrangements, can be located accurately by providing "quality of transmission" information during operation for every single node.

PROTECTION MEASURES

EtherCAT verifies whether a telegram was transmitted correctly and was processed correctly by all addressed devices. The standard Ethernet checksum found at the end of the Ethernet telegram is used for this purpose. Since one or several slaves modify the telegram during the transfer, the checksum is recalculated for each slave. If a checksum error is detected, a status bit is set in the EtherCAT slave, and an interrupt to the master is triggered if necessary, so that a fault can be located precisely.

During a read operation, the slave inserts the addressed data in the designated data field, from where they can be retrieved during writing. In both cases, the slave increments a working counter positioned at the end of each EtherCAT command. Since the master knows how many slaves are addressed by the telegram, it can detect from the counter whether all slaves have exchanged their data correctly.

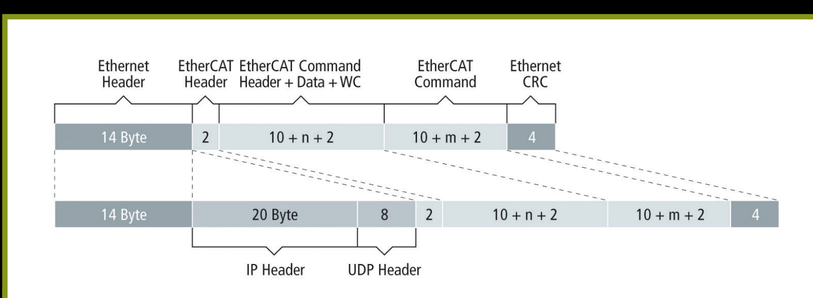
PROTOCOL

Fieldbuses have to meet different requirements in terms of data transmission characteristics. Parameter data are transferred acyclically and in large quantities, whereby the timing requirements are relatively non-critical. Diagnostic data are also transferred acyclically and event-driven, but the timing requirements are more demanding, and the transmission is usually triggered by a peripheral device. Process data, on the other hand, are transferred cyclically with different cycle times. It is important that 'dropped cycles' are avoided. The timing requirements are most stringent for process data communication.

EtherCAT has different addressing options for different types of communication, optimised for the particular requirements.

PEER-TO-PEER COMMUNICATION

Although EtherCAT uses a clear master/slave communication model, thereby ideally supporting hierarchical control technology, peer-to-peer (internode) communication between EtherCAT devices can be created very easily using the features of the Fieldbus Memory Management Unit (FMMU). To this end, memory areas from the 4GB logical address space are allocated for internode communication and cyclically exchanged by the master. The master alternately issues a read query and, in the next cycle, a write command for the respective memory area. All devices that are configured accordingly insert their peer-to-peer communication data or retrieve them during the next cycle. For the master, these data are transparent – it merely deals with the cyclic exchange.



This is the basic Ethernet telegram used by EtherCAT. For more complex switched systems, the UDP/IP stack may be added to route the message to the proper destination.

Compared with party line bus systems, in which all devices are connected to the same communication medium, one cycle is "wasted". However, this is more than compensated for by the outstanding usable data rate and the associated short cycle times. The strategy described above also has the advantage that the internode communication data are collected from several sources and then simultaneously arrive at all addressed destinations during the next cycle. At a cycle time of, for example, 100 microseconds, approximately 1000 bytes can be sent from almost any number of sources to the same number of destinations.

> ETHERCAT DOES NOT REQUIRE IP ADDRESSES – CONFIGURATION IS AUTOMATIC, CONTROLLED BY THE MASTER

DISTRIBUTED CLOCKS

In EtherCAT, distributed clocks enable all fieldbus devices to have the same time. One device is assigned the master clock, which is used to synchronise the control and the slave clocks of the other devices. The control sends a special telegram at certain intervals as required in order to avoid the slave clocks diverging beyond specified limits. In this special telegram the bus device with the master clock enters its current time. The slave clocks then read the time from the same telegram. The master clock must be located before the slave clocks in the ring.

ETHERCAT: A DIFFERENT ROUTE

In automation technology, there is currently a trend to bring Ethernet to the field level. Various approaches promise low cost, high bandwidth, simplified vertical integration, utilisation of standard components from the office, and low configuration and diagnostic effort – all this combined with the real-time capability.

At closer inspection however, many of the arguments become weak or change to the opposite: The comparatively high bandwidth of 100MBaud Ethernet is ruined if typical I/O nodes with few bytes of process data are used, each addressed by one telegram. A device with four bytes per direction, for example, achieves a usable data rate of 3-4%.

Higher costs tend to argue against use of Ethernet in the field. Apart from the pure connection costs, another factor is the relatively high computing capacity required for processing telegrams in the slaves. The use of standard components usually reaches its limit when a certain degree of real-time capability is required.

Furthermore, switched star cabling does not lend itself very well for use on the factory floor. The configuration is certainly not easier: allocation of IP addresses requires IT knowledge and tools, and frequently leads to conflicts with the IT department. Replacement of components with MAC IDs and IP addresses might be difficult.

EtherCAT takes a different route and combines the advantages of fieldbus technology with the otherwise indisputable advantages of the Ethernet world. The available bandwidth is almost fully utilised, and the costs are reduced to a simple ASIC connection in the EtherCAT device. Standard components are used where they are in fact standard – in the control and not in the 2 bit I/O terminal. EtherCAT does not require IP addresses, and configuration is automatic, controlled by the master using simple algorithms. Vertical integration is nevertheless available. Devices requiring an IP address can have one and are then integrated fully transparently in the network.

EtherCAT enables high-performance machine controls to be realised, capable to exchange many distributed signals with cycle times significantly below 100µs. Moreover, the system is just as suitable for cost-effective control applications where cycle times three orders of magnitude larger are sufficient, e.g. building automation with 100ms. In this case, any commercially available PC or any controller with integrated Ethernet port can be used as a master. EtherCAT therefore offers a unified, powerful communication basis for the entire automation sector. The same system technology can be used from "small" PLCs for less than €100 to high-performance CNC devices. ■

Dr Dirk Jansen is head of Software Development, Lead System, CNC and I/0, and Holger Büttner is Manager of the Berlin Engineering Office. Both are with Beckhoff Germany. For more information about EtherCAT readers may visit www.ethercat.org