Tasty solutions for real-time Ethernet control and device connections

Various organizations have developed protocols for using Ethernet in process-related automation technology. These protocols address different areas of implementation, which will result in a variety of solutions becoming established on the market. Automation technology providers will therefore have to ensure that their devices support several Ethernet-based real-time communication systems in the future. Based on the hyNet processor family and the corresponding operating systems and protocol implementations, it is possible to create solutions which are flexible, economical and powerful.

1. Fulfilling the requirements for real-time behavior in various real-time Ethernet protocols

1.1 Issues

Because they are so widespread, Ethernet and the IT protocols associated with it (TCP/UDP/IP, HTTP, …) allow for the technologically unlimited exchange of data. Information can be retrieved and transmitted from anywhere. The significance of this factor is growing in the field of automation technology, as access to information becomes more and more important. For this reason, many companies and organizations have become involved in projects aimed at increasing the use of Ethernet in industrial communication.

The Ethernet protocol alone, however, cannot fulfill the requirements for real-time behavior and determinism in automation technology. It has been shown that when a certain communication load has been reached, data transmission is no longer possible because collisions arise. Sufficient bandwidth cannot always be guaranteed under all conditions. This has been identified as a restriction in IT as well. Collisions can be prevented by employing switches and full-duplex transmission. Since all data packets must pass through the switch, it creates a new bottleneck. By prioritizing the data packets, it is possible to determine the maximum delay the telegrams in the various priority classes will experience. When the communication load in a system is known, a certain amount of bandwidth can be guaranteed for various applications. By definition, a telegram which has begun to be forwarded by a

switch cannot be interrupted if a higher-priority telegram arrives. This second telegram must wait to be forwarded. The waiting period depends on whether a low-priority telegram is being sent, when the transmission began and how long the telegram is. This gives rise to a jitter, since there is no guarantee that telegrams sent cyclically will be forwarded through the switch in the same cycle.

The various real-time Ethernet systems (RTE) take different approaches to achieving short cycle times and minimal jitter while remaining compatible with Ethernet technology.

1.2 PROFINET IO approach

PROFINET IO always requires the use of switches and full-duplex transmission. Since the protocol supports telegram prioritization, it is possible to achieve very short cycle times (2-50 ms depending on the volume of communication). PROFINET IO plans the communication process to optimize the utilization of the communication channel and ensure the jitter remains small. Optimal send and receive times for the relevant data packets are calculated by taking all system conditions into account (amount of real-time data to transmit cyclically, proportion of other data, delays at the switch and on the lines). This information is loaded in the devices. This approach demands certain features from the project engineering tool. The devices used must have a highly precise distributed time base so that frames can be sent and received in accordance with specifications. The time synchronization is based on IEEE 1588. This approach also requires that the switch functionality be expanded.

1.3 EtherNet/IP approach

EtherNet/IP also requires the use of switches and full-duplex transmission. With EtherNet/IP, the frames to be transmitted are assigned to various types of messages with regard to the required determinism, response time and error detection. The protocol defines measures (prioritization of frames, direct use of Ethernet instead of UDP/IP, etc.) for fulfilling the respective requirements. With CIPsync, each application variable is given a valid execution date. If the date arrives to late, it is ignored. A highly precise distributed time base is also necessary for assigning time stamps and making decisions as regards timeliness. EtherNet/IP also uses IEEE 1588 here. The process of calculating the optimal utilization of the system places high demands on the project engineering tool.

1.4 ETHERNET Powerlink approach

ETHERNET Powerlink relies on hubs. To achieve short cycle times and minimal jitter, a corresponding protocol is used on Ethernet. This prevents the collisions which may arise when hubs are used. The protocol demands short response times from devices (controlled nodes) and high communication performance from masters (managing nodes) when sending and processing frames.

1.5 EtherCAT approach

Neither switches nor hubs are required here. An Ethernet frame is sent by a master and passes through the various slaves. When passing through the slaves, data is extracted and inserted on the fly. No protocol processing takes place as such. Instead, the respective device "knows" where input and output data can be found. Since this functionality is realized in hardware, very short throughput times can be achieved by the devices.

2. Requirements of a combined hardware and software platform

With PROFINET IO, ETHERNET Powerlink and EtherNet/IP, there are no special hardware requirements for realizing simple real-time solutions. It is enough to efficiently implement the protocol in a software stack. EtherCAT, however, is optimized for high demands and will not work without hardware support. This also applies to complex solutions with PROFINET IO, ETHERNET Powerlink and EtherNet/IP. Hardware support is necessary for realizing the IEEE 1588 standard and for processing the protocol.

2.1 Hardware support for the IEEE 1588 protocol

This standard describes a protocol for synchronizing a distributed time base. Use of the protocol should minimize the deviation between clocks on various devices in a system. The protocol is quite simple: There are frames for determining the line delay between participants and synchronization messages. Based on this delay, participants can determine the amount of deviation between their clock and the master time and make the appropriate adjustments. The precision of the procedure depends both on the quality of the local clock generation and on the jitter-free calculation of delays. Jitter can arise when non-deterministic components (software) are registered while the delay is being calculated. The closer the assignment and recording of time stamps is to the transmission medium, the more precise the clock synchronization will be.

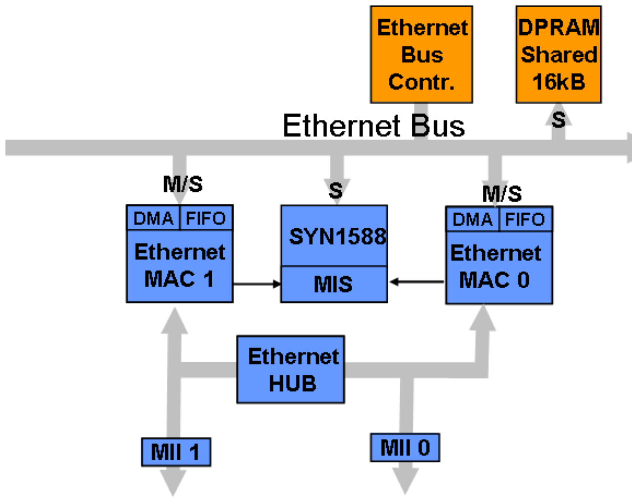
Although IEEE 1588 is an international standard, very few Ethernet connections currently support this protocol. Implementing it therefore requires special hardware.

2.2 Hardware support for processing the respective RTE protocol

The achievable data throughput using a particular communication medium is only one criterion for evaluating the performance of a system. For complete evaluation, one must also consider the time it takes the data to pass through the protocol stack. This can be optimized with the appropriate hardware. EtherCAT is an extreme example, since the entire protocol is processed in hardware in the slave. With ETHERNET Powerlink, too, the required response times cannot be achieved without hardware support. With EtherNet/IP and PROFINET IO, many other frames – besides just the protocols relevant to real-time processing – may arise and be taken into account by the protocol software. Pre-selection on the basis of hardware support relieves the strain on the protocol software and improves response times.

3. Support for implementing real-time Ethernet protocols in hyNet processors

The image shows the Ethernet subsystem of the hyNet XS processor. All hyNet processors have already realized IEEE 1588 in hardware. The corresponding function block is a part of the subsystem. The MII scanner (MIS) checks telegrams which arrive in the MAC for their relevance to IEEE 1588. As mentioned earlier, there are protocol elements for determining the line delay and protocol elements with which synchronization messages are exchanged. IEEE 1588 does not define the network or transport protocol on which these protocol elements are transmitted. It could be UDP; with EtherNet/IP and PROFINET IO, it is Ethernet. These conditions can be programmed in the MIS (MII scanner). Regardless of how the protocol is implemented, this guarantees that time stamps are recorded and assigned as closely as possible to the pure communication medium.



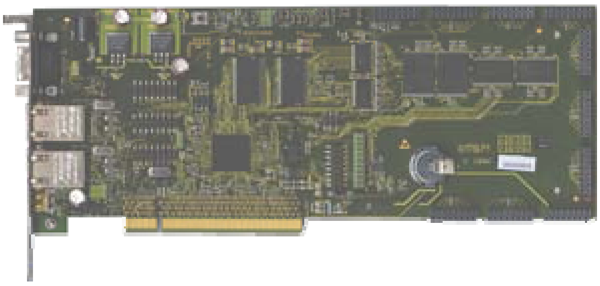
To ensure the optimal transfer of data, every MAC component can use an integrated DMA controller for sending and receiving. The data transfer is controlled by descriptors which are set using software. The descriptors contain, amongst other things, storage addresses for the telegrams, status and control information, and an address for the next descriptor.

This makes it possible to create list structures for the data traffic. Frame pre-processing can also be controlled through the MIS.

The hub functionality needed for ETHERNET Powerlink is already integrated in the chip as well. The internal delay through the hub, measured between the MII, is 120 ns.

4. Complete solution

The PROFINET IO protocol software is available for hyNet processors. An Evaluation Kit acts as a test and development platform for an effective introduction to the technology. In addition to the evaluation board pictured in Image 3, it also includes the PROFINET IO protocol stack, a free development environment, commissioning support and example programs. Other RTE protocols will follow.



In addition to Ethernet functionality, the hyNet processors have a number of other features which make them ideal for use in compact automation devices. These include extensive peripheral functions (PWM, serial interfaces, support for video processing and much more).

The authors:
Frank Iwanitz, Softing AG
Axel Mehnert, Hyperstone AG