

The Time-Triggered Communication Protocol TTP™/C

Hard real-time and composability are requirements of increasing importance in the embedded systems market. Conventional event-triggered communication protocols, such as CAN, are appropriate for soft real-time systems that require flexibility and do not have substantial timeliness and dependability requirements. If composability, hard real-time performance and dependability are more prominent issues than flexibility, then the Time-Triggered Protocol TTP™/C is most suitable. TTP™/C provides clock synchronization, hard real-time message delivery with minimal jitter and very comprehensive error detection mechanisms. In order to fully exploit these advantages, a development environment for TTP™/C-based systems following a two-level design approach has been developed by TTTech.

THE INCREASING IMPORTANCE OF FAULT-TOLERANCE

In many embedded real-time applications, such as automotive, computer safety is currently approached at two levels: At a basic level, a mechanical system provides the degree of safety that is considered sufficient for safe operation. On top of this basic mechanical system, a computer system provides optimized performance. In case the computer system fails, the mechanical system takes over. An anti-lock braking system (ABS) is a typical example of this approach: if the computer fails, the conventional mechanical brake is still operational. In the near future, this approach to safety will reach its limit for two reasons:

1. The improved price/performance of the microelectronic components will make the implementation of fault-tolerant computer systems cheaper than the implementation of mixed (computer/mechanical) systems. As a consequence, there will be a cost pressure to eliminate the redundant mechanical system.
2. As the performance of the computer controlled system is further improved, the fall-back to the inferior performance of the mechanical system increasingly constitutes a safety risk for the operator who is

accustomed to the high performance of the computer controlled system.

Both trends favor the deployment of fault-tolerant real-time systems that will provide the specified service despite a failure of any one of their components. Until today, the aircraft domain has been the technology leader in the field of fault-tolerant real-time systems. A well-known example is the flight control system of the Airbus A320. Time-triggered systems currently used in the aircraft domain share some ideas with TTP™, but are too costly to be applied outside the aerospace market. Dependable real-time performance, however, is an issue in many cost-sensitive industry sectors such as cars, industrial control, public transportation, robotics, medical electronics or building control. The cost advantage of TTP™-based solutions makes time-triggered systems accessible to this wide variety of industries.

COMPOSABILITY PROBLEMS: UNMANAGED COMPLEXITY IN LARGE DISTRIBUTED SYSTEMS

One of the most difficult issues in the design and implementation of dependable real-time systems is the seamless integration of heterogeneous electronic subsystems-possibly developed by different suppliers-

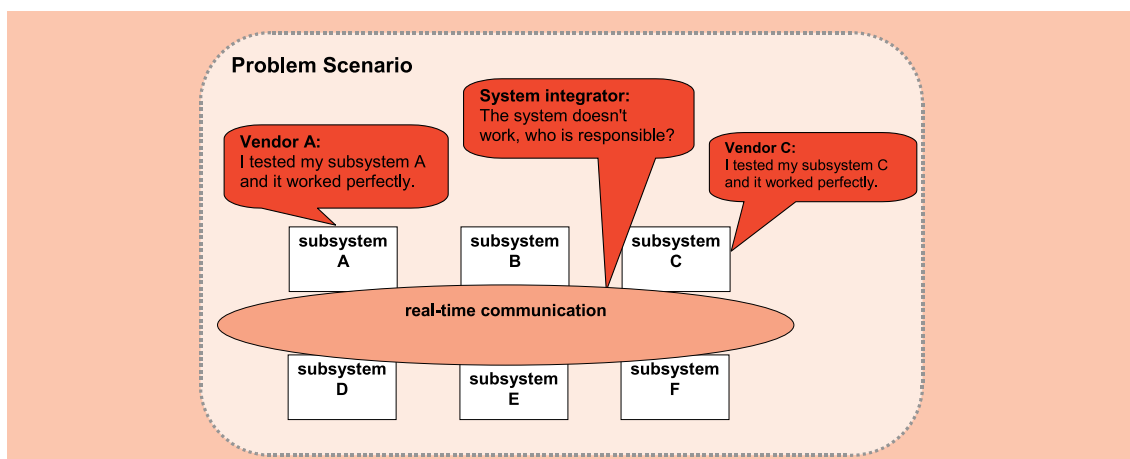


Figure 1: Composability problem scenario.

Ad
National Instruments

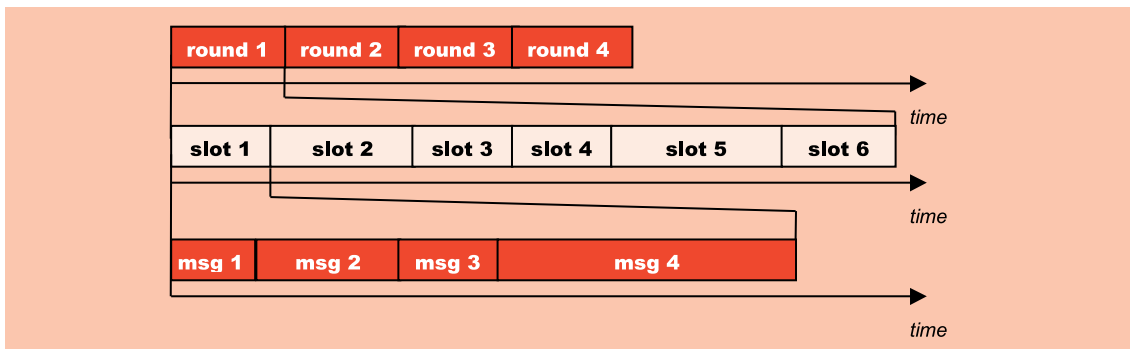


Figure 3: TDMA bus access strategy.

into an integrated computer system.

An architecture is said to be composable with respect to a certain property if system integration will not invalidate this property once the property has been established at the subsystem level. For hard real-time systems composable timing behavior is the most critical issue that cannot be guaranteed with even-triggered communication systems.

In the future, the replacement of conventional subsystems by a multitude of electronic by-wire systems built by competing suppliers will aggravate this problem. Consequently, system integrators have a pressing need for a composable network architecture providing clear separation between the system- and subsystem issues. This requires a communication system with well-defined interface in the value and time domain.

THE TIME-TRIGGERED COMMUNICATION PROTOCOL TTP™/C

TTP™/C is an integrated communication protocol for hard real-time fault-tolerant distributed systems. TTP™/C is member of the TTP™ protocol family where C indicates that it satisfies SAE (Society of Automotive Engineers) Class C requirements for hard real-time fault-tolerant communication the automotive area.

TTP™/C provides hard real-time message delivery with minimal jitter. Different fault-tolerance strategies are supported. It guaranteed that no single failure of any part of the communication system could lead to a disruption of the communication. TTP™/C provides a dis-

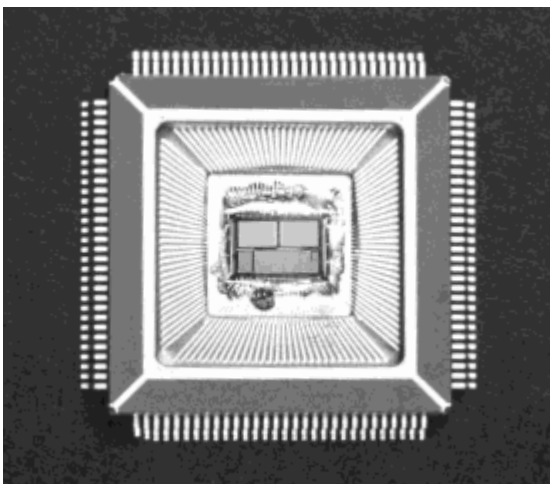


Figure 2: TTech's silicon implementation of TTP/C™.

tributed fault-tolerant clock synchronization. Extensive mechanisms for error detection, recovery and re-integration of nodes are provided. The protocol has been designed for highest data efficiency and minimal protocol overhead. Furthermore, TTP™/C supports compositability by its precisely defined behavior in the value and time domain.

CLOCK SYNCHRONIZATION AND TDMA CHANNEL ACCESS

The time-triggered protocol TTP™/C is based on time as its underlying driving force, i.e., all activities of a system are carried out in response to the passage of certain points in time. It is therefore necessary that all nodes in the system have a common notion of time. This common notion of time is provided by the TTP/C communication protocol, which is based on fault-tolerant clock synchronization. The current TTP™/C silicon controller implementation provides a synchronized clock with 1 μ s tick duration. It is therefore possible to carry out globally synchronized actions or to implement distributed control loops with minimal jitter.

TTP™/C is realized based on the TDMA (time division multiple access) bus access strategy. The TDMA bus access strategy is based on the principle that the individual communication controllers on the bus have time slots allocated where exactly one communication controller is allowed to send information on the bus. It is thus possible to predict the latency of all messages on the bus, which guarantees hard real-time message delivery. Furthermore, since the messages are sent at an a priori pre-determined point in time the latency jitter is minimal. The clock synchronization is based on the TDMA principle as well. All nodes in the cluster know when a certain node has to send a message. By comparing this a priori know point in time with the actual time when the message is received, each receiving node knows the difference between the sender's clock and its own clock. This information is used to implement clock-synchronization without sending any overhead messages.

FAULT-TOLERANCE

Another very important feature of TTP™/C is its ability to guarantee that no single failure of a node can disturb the communication of the remaining nodes in the cluster. In a conventional event-triggered system a faulty node-called babbling idiot-is able to monopolize the communication media by sending high priority

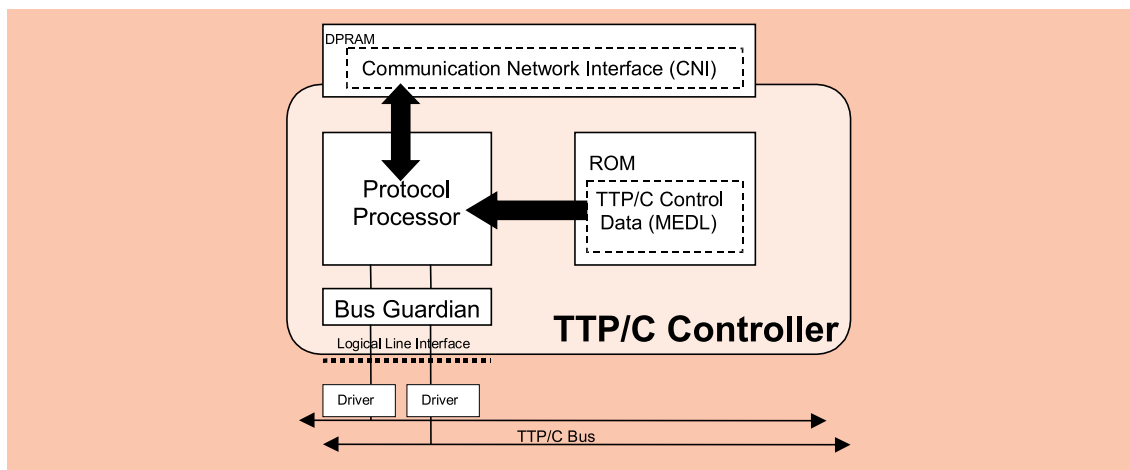


Figure 4: Structure of the TTP™/C communication controller.

messages permanently. This single point of failure effectively prevents correct nodes from exchanging messages. With TTP™/C this failure the so-called bus guardian who is a separate block of the TTP™/C communication controller prevents mode. Since the bus access strategy is TDMA a certain node is not allowed to access the bus outside its pre-allocated time window. This is ensured by the bus guardian in such a way that the permission to transmit is given only during the nodes send time window. Therefore, no single node can disturb the communication of the other nodes on the network.

Furthermore, TTP™/C provides two serial communication channels. This redundancy can be utilized for fault-tolerance to ensure that critical messages arrive even in the presence of a broken channel or to double the throughput. The selection between fault-tolerance and throughput can be taken on a per message basis.

Another important feature of TTP™/C easing programming of fault-tolerant applications is its replica determinism: a message arrives at all recipients at exactly the same time with exactly the same contents or it does not arrive at all. The application software can therefore process a message without having to worry that other nodes receive the message at a later point in time or with different arrival order.

AUTONOMY AND COMPOSABILITY

The time-triggered communication paradigm of TTP™/C allows making the communication controller completely autonomous. In an event-triggered system the host computer has to initiate all send operations and has to react on all received messages. A communication controller in event-triggered system therefore processes the commands of the host CPU without autonomy and tight temporal coupling. With TTP™/C the communication controller has its own memory and data structures—the MEDL (message descriptor list)—that defines all the protocol activities such as the points in time when to send and receive messages. The temporal coupling between the communication controller and the host is thus restricted to a minimum. The host CPU can read and write messages from the communication controller via a simple dual port RAM interface. The TTP™/C communication controller sends and

receives all messages completely autonomous, it does not require any commands from the host. This gives ideal support for composability: The communication pattern or timing on the bus can be defined precisely for a complete cluster. The resulting MEDLs are loaded into the communication controllers. This clean separation between communication and application software precludes all sort of timing failures in the application software leading to communication failures. It becomes much easier to integrate different subsystems or nodes since there is no variance or uncertainty in the communication timing. Problems as depicted in figure 1 cannot occur.

ERROR DETECTION

While conventional event-triggered protocols provide error detection on the sender's side only TTP™/C is able to provide error detection at the receiver's side. This is possible because all communication partners in a cluster know a priori the points in time when a certain message has to be sent. Based on the information about missing messages the receiver becomes autonomous and can decide on its own how to react on the fact that the sender is no longer providing this information.

A second very strong service for error detection is the membership service of TTP™/C. The membership service gives all correct nodes a consistent and timely view on the status (correct/fail) of all the nodes in the cluster. This service supports the application software by giving a consistent view on the systems state and degree of functionality.

IMPROVED COMPOSABILITY THROUGH A TWO-LEVEL DESIGN FRAMEWORK

From the above said it follows that a development environment for TTP™-based systems must support the precise specification of the temporal and functional interfaces between the subsystems of a time-triggered architecture: At the system-level, a system integrator (e.g., an automotive company) defines the subsystem functions and specifies the communication interfaces in the value and time domains precisely. At the subsystem-level, the component supplier retains complete

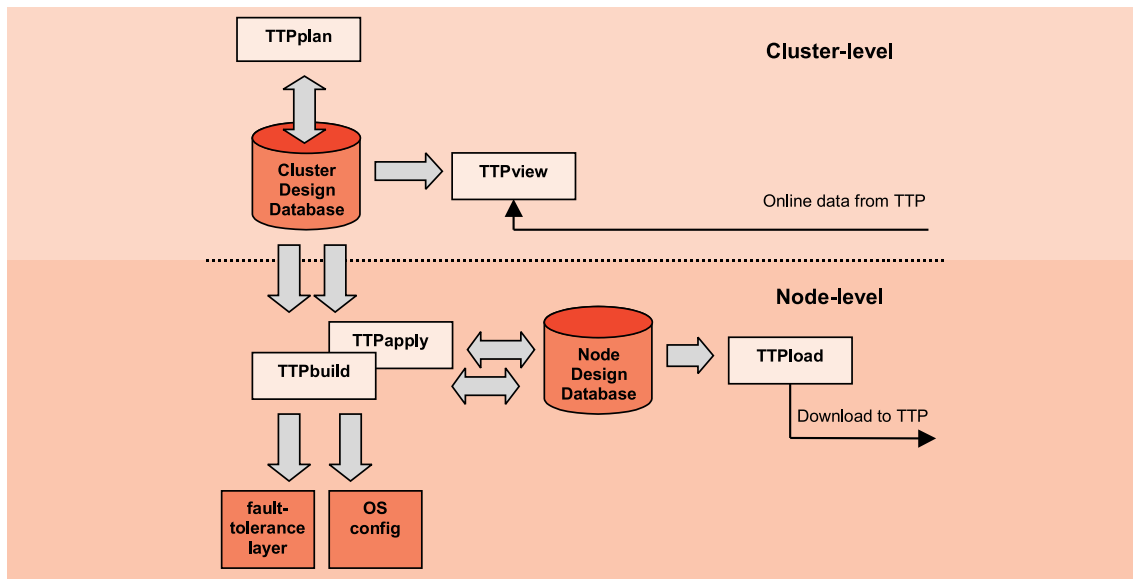


Figure 5: TTEch's development environment for TTP™-based systems.

control over all hardware and software design decisions as long as he complies with these interfaces. The unambiguous interface specification enhances ease of integration, quality, and reusability of the developed products. The composability of the architecture prevents the occurrence of unintended integration effects during system integration.

THE BENEFITS OF A TWO-LEVEL DESIGN FRAMEWORK

The separation of concerns inherent to this two-level design framework translates directly into significant business benefits for all parties involved. The clear definition of responsibilities prevents the omission of essential functions as well as the duplication of efforts resulting in waste and potentially conflicting implementations.

For both system integrator and subsystem suppliers, the delimitation of responsibilities restricts the potential for conflicts and reduces the communication overhead.

For the subsystem supplier, the two-level design framework affords a high degree of independence. The supplier is unrestricted in his design choices, needs less system information, and benefits from a substantial decrease in requirements changes.

For the system integrator, the two-level design framework offers all the benefits of composability: the costs, time, and risk of system integration are reduced and the need for cost-intensive substitutes for temporal composability alleviated. By supporting faster and more precisely estimable turn-around times, the approach permits a shorter time-to-market.

Figure 5 shows TTEch's software development environment that implements the described two level design framework. It includes a cluster design tool (TTPplan™) and a monitoring tool (TTPview™) on the cluster-level, as well as node-level tools for downloading (TTPload™), for adapting the bus schedule to node specific properties (TTPapply™), for fault-tolerance layer generation and operation system configuration

(TTPbuild™).

CONCLUSION

In order to satisfy hard real-time and composability requirements, an innovative multiplexing protocol is required. Deriving temporal control signals from the progression of physical time, the time-triggered communication protocol TTP™ tackles the problem of system predictability at its root. A software development environment systematically tailored to the properties of TTP™-based systems is presented. ■

PREFERENCES

- H. Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers. 1997. ISBN 0-7923-9894-7.
- H. Kopetz and G. Grünsteidl. TTP—A Protocol for Fault-Tolerant Real-Time Systems. IEEE Computer, January 1994, pp. 14-23. 1994.
- S. Poledna. Fault-Tolerant Real-Time Systems: The Problem of Replica Determinism. Kluwer Academic Publishers. 1996. ISBN 0-7923-9657-X.
- H. Kopetz. A Comparison of CAN and TTP. Proceedings of the IFAC Distributed Computer Systems Workshop, Como, Italy 1998.

Stefan Poledna received the M.Sc. and Ph.D. degrees in computer science from the Vienna University of Technology, Austria, in 1991 and 1994, respectively. He has more than 15 years experience in the automotive industry. Dr. Poledna lectures Fault-Tolerant Computing Systems at the Vienna University of Technology. Since its foundation in 1998 he is managing director of TTEch Computertechnik GmbH.

Georg Kroiss is currently finishing his M.Sc. at the Vienna University of Economics and Business. He is specialized in the field of innovation and high-tech marketing. Mr. Kroiss is member of the High-Tech Marketing group and is working in joint projects with TTEch.