Time Triggered Communication on CAN (Time Triggered CAN- TTCAN)

Thomas Führer
Bernd Müller
Werner Dieterle
Florian Hartwich
Robert Hugel
Michael Walther
Robert Bosch GmbH

Connecting microcontrollers, sensors and actuators by several communication systems is state of the art within the electronic architectures of modern vehicles. Today's communication is widely based on the event triggered communication on the Controller-Area-Network (CAN) communication protocol. The arbitrating mechanism of this protocol ensures, that all messages are transferred according to the priority of their identifiers and the message with the highest priority will not be disturbed. In the future some mission critical subnetworks within the upcoming generations of vehicle systems, e.g., x-by-wire systems (xbws), will require additionally deterministic behavior in communication during service. Even at maximum bus load, the transmission of all safety related messages must be guaranteed. Moreover it must be possible to determine the point of time when the message will be transmitted with high precision.
One way to solve this issue using CAN is the extension of the standard CAN protocol to a time triggered protocol TTCAN. The communication is based on the periodic transmission of a reference message by a time master. This allows to introduce a system wide global network time with high precision. Based on this time the different messages are assigned to time windows within a basic cycle. A big advantage of TTCAN compared to classic scheduled systems is the possibility to transmit event triggered messages in certain "arbitrating" time windows as well. These time windows, where normal arbitration takes place, allow the transmission of spontaneous messages. This paper describes this extension "TTCAN" as it is accepted in TC22/SC3/WG1/TF6 (ISO11898-4) as the common base for the standardization work.

1. Future system architectures and the communication system.

Connecting microcontrollers, sensors and actuators by a communication system becomes more and more popular within the electronic architectures of modern vehicles. Synergy effects by distributed application functionality via several microcontrollers and the use of sensor information over the network leads to more complex system architectures with many different subnetworks running on different speed and different protocol implementations. Networking even reaches such a state of complexity that a careful system design becomes necessary. Unintended side-effects via the network on the microcontrollers and on the application have to be avoided during runtime.
To overcome the increasing complexity of these systems a deterministic behavior of the communication network must be provided, in particular when dealing with distributed functionality or redundant realization of certain nodes. This can be achieved by using the design philosophy of time triggered operation [1] at the communication network (time triggered protocol activities [2]) and at the application level (time triggered task activation [3]). The optimum is reached if a globally synchronized time base (global time) is available at all nodes of the network with a precision which fulfills the real-time requirements of the application.

Several systems will require this behavior of the communication system, e.g. x-by-wire systems (xbws), motor management systems and sensor subnetworks. Within the automotive domain, xbws will be one of the most mission critical systems. They control the vehicle and its dynamics while the input from the driver, e.g. steering wheel, to the system is mechanically and/or hydraulically de-coupled from the physical transaction on the road, e.g. road wheels. The communication network is the back-bone of these by-wire applications [4] and often has to be redundant as well. The first generations of xbws will still remain with mechanical/hydraulic backup, but determinism of the message transfer, the service of a global network time and redundancy at the timing services will already be important for these first generation xbws.

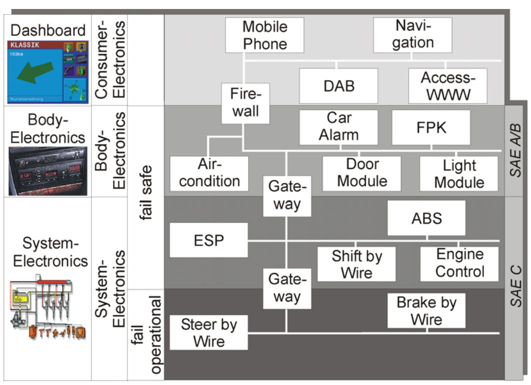

Figure 1 - Overview of different systems and networks within future electronic architectures.

Figure 1 gives an overview of some of the applications in a vehicle which may be connected via subnetworks. The subnetworks are optimized for their purpose, so the multimedia will need a very high bandwidth while the x-by-wire network will emphasize fault-tolerance and safety.
But also several applications from other industry domains, e.g. automation industry or medical service industry build many software solutions based on CAN and will welcome the additional features of TTCAN improving the determinism of their networks.

2. Time Triggered Operation

Purely time triggered operation of the communication system means that any activity is determined by the progression of a (globally synchronized) time. Sending, receiving or any other activity, depends on a predefined time schedule and on the current state of the clock as it is shown in Figure 2. Message A is sent if the system clock reaches 3 and 6 while message C is sent at 5. If the whole communication traffic is summarized in such a time table, a deterministic and predictable communication matrix results. The necessary information can be mapped into each node within the network. This results in a highly composable system in the domain of time and value [5].

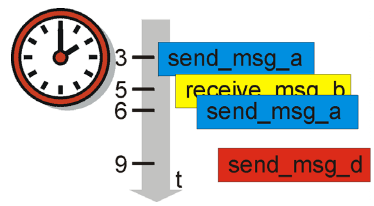

Figure 2 - Time triggered operation of the communication system – sending time triggered messages.

3. Time-Triggered operation on CAN

One of the most powerful features of the CAN protocol [6, 7] is the bitwise arbitration to control the media access among the controllers of the network. The bitwise arbitration guarantees a controller with a high priority message to access the bus even if other controllers try to access the media without destroying this message. The access may be delayed if some other message is already in the process of transmission or if another message with higher priority also competes for the bus. This means that even the temporal behavior of the message with the highest priority may show a small latency. The lower the priority of a message is, the higher the latency jitter for the media access may be [9].
The goal of time triggered operation on CAN is to avoid these latency jitters and to guarantee a deterministic communication pattern on the bus. Moreover, this allows to use the physical bandwidth of a CAN network much more efficiently (under the constraint of determinism). The well known protocol specification of ISO11898 [6, 7] is extended for time triggered execution of CAN within ISO11898-4 in two levels. Extension level 1 guarantees the time triggered operation of CAN based on the ref-

erence message of a time master. Fault-tolerance of that functionality is established by redundant time masters – the so called potential time masters. In extension level 2 a globally synchronized time base is established and a continuous drift correction among the CAN controllers is realized.

3.1 The Reference Message

TTCAN is based on a time triggered and periodic communication which is clocked by a time master's reference message. The reference message can be easily recognized by its identifier. Within TTCAN's level 1 the reference message only holds some control information of one byte, the rest of a CAN message can be used for data transfer. In extension level 2, the reference message holds additional control information, e.g. the global time information of the current TTCAN time master. The reference message of level 2 covers 4 bytes while downwards compatibility is guaranteed. The remaining 4 bytes are open for data communication as well.

3.2 The basic cycle and its time windows

The period between two consecutive reference messages is called the basic cycle (see Figure 3). A basic cycle consists of several time windows of different size and offers the necessary space for the messages to be transmitted.

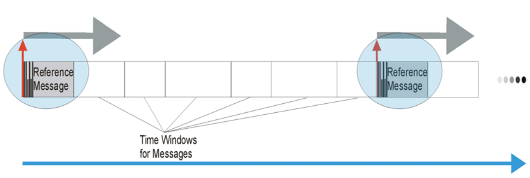

Figure 3 – The reference message starts the TTCAN basic cycle.

The time windows of a basic cycle can be used for periodic state messages and for spontaneous state and event messages. Any message that is sent has the CAN data format and is a standard CAN message. A time window for periodic messages is called an exclusive time window. Within exclusive time windows the beginning of the time window determines the sending point of a predefined message of a node. If the system was properly specified and the off-line design tool analyzed the communication pattern, no conflicts will happen. However, even in the error case of a conflict, the CAN protocol properties (bit arbitration, only sending when the bus is idle) are valid. The system engineer has to decide off-line which message must be sent at which exclusive time window. To provide higher flexibility to the system designer, an exclusive time window may be repeated more than once within a basic cycle. The automatic retransmission of CAN messages is not allowed in exclusive time windows.
A time window for spontaneous messages is called an arbitrating time window. Within an arbitrating time window the bitwise arbitration decides which message of which node in the TTCAN network will succeed (see Figure 4) on the bus. At design time it is allowed to schedule more than one message for an arbitrating time window. So the application can decide at runtime if it would like to use an arbitrating window for a message to be sent and which message should be sent in a certain arbitrating window. The automatic retransmission of CAN messages is also not allowed within arbitrating time windows.

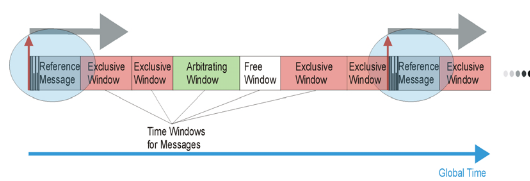

Figure 4 - Exclusive time windows and arbitrating time windows of a TTCAN basic cycle.

During the design phase it is also possible to reserve free time windows for further extensions of the network. They can be changed to arbitrating or exclusive time windows if new nodes need further space for communication or the bandwidth has to be extended for existing nodes.

3.3 The node specific knowledge in TTCAN

In TTCAN a network controller does not have to know all messages of the network. The controller only gets the necessary information it needs for time triggered sending and receiving of messages as well as for sending of spontaneous messages. An example is shown in Figure 5 where the controller sends message C in the exclusive time windows 2 and 6 and sends the spontaneous message F in the arbitrating time window 3. The controller is only interested on the reception of message A in the exclusive time window 1. The node specific knowledge in TTCAN is kept at a minimum compared to other strictly time triggered communication systems.

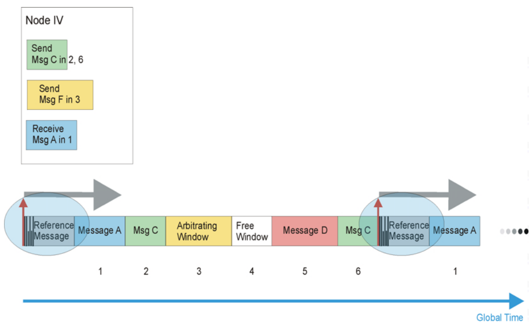

Figure 5 - TTCAN communication - Local information of a TTCAN controller.

This design principle allows highly optimized memory utilization in a hardware realization but offers still enough information for network management, e.g. within OSEKTime's FTCOM [3]. Moreover, this provides a very high flexibility during development as changes in the schedule imply a new download of the schedule only for the affected controllers.

3.4 The System Matrix

Practice has shown that applications include many control loops and tasks with different periods. They all need individual sending patterns for their information.

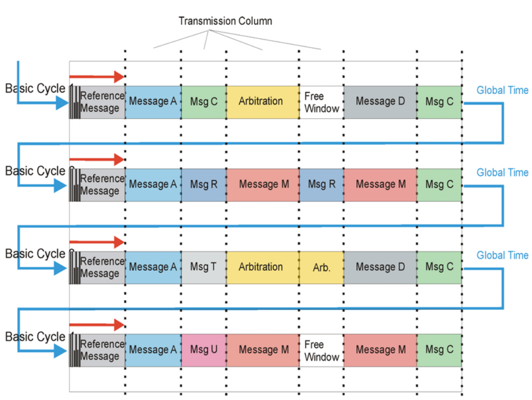

Figure 6 - Example of a TTCAN system matrix (several basic cycles build the so called matrix cycle)

The TTCAN basic cycle would not offer enough flexibility to satisfy this need. The TTCAN specification allows to use more than one basic cycle to build the communication matrix or system matrix of the systems engineer's needs. Several basic cycles are connected to build the matrix cycle. Most patterns are possible, e.g. sending every basic cycle, sending every second basic cycle, or sending only once within the whole system matrix. An example is shown in Figure 6.

TTCAN specification allows also another useful exception. As it is shown in Figure 6 the system matrix is highly column oriented. It may make sense to ignore the columns in the case of two or more arbitrating time windows in series. This principle is shown in Figure 7.

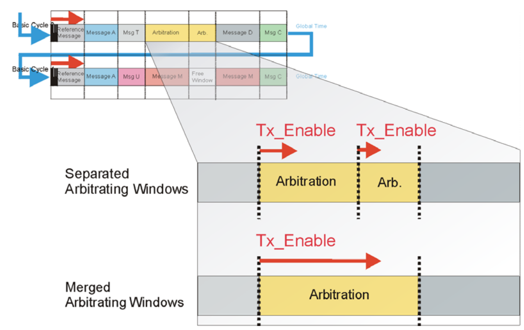

Figure 7 - Merged arbitrating windows

The most important constraint for this construct is that the starting point of a spontaneous message within this merged arbitrating window is not allowed if it will not fit in the remaining time window. The start of the next periodic time window must be guaranteed. This is the task of an off-line design tool used to build TTCAN system matrices.

The automatic retransmission within a merged arbitrating time window is allowed as long as the constraint already described above is satisfied.

3.5 Sending and receiving of messages using time marks

Within a basic cycle of TTCAN, the protocol execution is driven by the progression of time. This time is the so called cycle time of TTCAN and is restarted after the reception of every reference message. The necessary link between the cycle time and the system matrix are the so called time marks. They specify the beginning of the exclusive and arbitrating time windows. Time marks for sending periodic or spontaneous messages are called TxTriggers. RxTriggers have to be defined to check the reception of a periodic message.

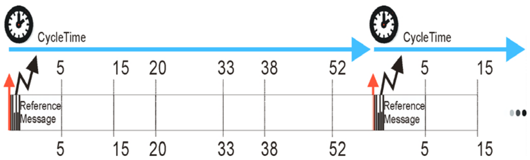

Figure 8 - TTCAN's cycle time and the time marks

A time mark furthermore consists of the base mark and the repeat count information. The base mark determines the number of the first basic cycle after the beginning of the matrix cycle in which the message must be sent/received. The repeat count determines the number of basic cycles between two successive transmissions/receptions of the message (see Figure 6).

3.6 Generation of the network time unit (NTU) in TTCAN

The cycle time of TTCAN is the basic time to guarantee the time triggered operation of the protocol. An important property of such a time information is the granularity of the time. The granularity of any timing information within TTCAN is the network time unit (NTU). So the cycle time is measured in NTU and is based on the nominal CAN bit time in TTCAN level 1 (see introduction of chapter 2) and on the physical second (2^{-n}) in TTCAN level 2. In level 2, to establish a system wide NTU, the node local relation between the physical oscillator of a TTCAN controller and the system wide NTU has to be established.

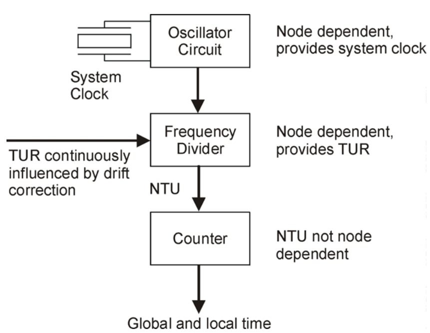

Figure 9 - Generation of NTU and the influence of TUR

Figure 9 demonstrates the principle of the NTU generation. The node dependent oscillator circuit provides the system clock to a frequency divider. This frequency divider generates the system wide NTU while a node local time unit ratio (TUR) takes care for the correct relation between the system clock and NTU. NTU now can be used to build a local time and to build the global time.

3.7 Generation of the global time and drift correction in TTCAN level 2

The node sending the reference message is the time master of the TTCAN network. In TTCAN level 2 all nodes take a snapshot of their time values at the frame synchronization pulse (e.g. sample point of SoF bit of reference message). The time master sends its (by definition correct) global time value for this frame synchronization pulse as part of the reference message. After reception each node can build its local offset as the difference between the master global time snapshot value and the own local time snapshot value. During the next basic cycle the node can compute the global time by global time = local time + local offset. If local time and global time have the same speed this ensures that all nodes have a consistent view on the global time. Due to slightly different clock drifts of the different nodes a mechanism has to be introduced to guarantee that local and global time have in fact the same speed. This mechanism is the continuous update of TUR. An initial value of TUR is a priori known node locally by the oscillator specification. During operation, to adapt this value to the correct value determined by the master clock speed the node measures the length between two successive frame synchronization pulses both locally (number of oscillator periods in this interval) and in global time (difference between the two master snapshot values). The quotient of these two values gives the actual TUR (limited only by the precision of the measurement). The achievable precision determines a reasonable [10] choice of the NTU-value in physical seconds. In level 2 the global time values of two nodes will then not differ by more than one NTU. The NTU typically will be in the order of a CAN bit-time.

4. Fault-Tolerance of the TTCAN time master

As the time master plays a vital role within the TTCAN approach, fault-tolerance of this functionality must be established. This is done by predefining more than one TTCAN controller to be a potential time master. They get corresponding identifier which can be used to control the startup behavior. After reset of a TTCAN controller, a potential time master checks if there is already traffic on the bus and if there is already a reference message sent. If not the potential time master sends a reference message with its identifier and, in level 2, with its local time as the very first global time of the network. This TTCAN controllers assumes to be time master. Whenever a reference message with a higher priority is received the potential time master stops sending the reference message and synchronizes to the basic cycle given by the higher priority time master. Whenever a reference message with a lower priority is received, the potential time master first synchronizes to the existing basic cycle and then tries to become time master by sending its own reference message at the start of the next basic cycle. Due to higher priority it will win the arbitration. Hence the protocol mechanisms ensure that out of all error free potential time masters the one with the highest priority eventually becomes active time master without violating the structure of the basic cycles. This state then is stable as long as no errors occur.

During operation a missing reference message is recognized by all potential time masters within short latency. The latency is realized by a timeout. After this timeout is reached a potential time master starts sending the reference message with its global time (local time + local offset) as content. The functionality of the time master is reestablished and the reference message is sent. Again, the bitwise arbitration of the standard CAN protocol decides among competing potential time masters.

5. Event synchronized initialization of a basic cycle

For some applications a strictly time triggered schedule over a long period is not the perfect solution. Relative to some events guaranteed latency times must be provided, however these events themselves not necessarily occur periodic in time. An example for such a requirement may be the motor management system where several tasks highly depend on the current position of the crank shaft. But as soon as a certain position is reached a predefined deterministic behavior is necessary, e.g. reading of sensor values via the communication bus. For these applications a strictly periodic schedule makes sense for some time, but it is important to be able to interrupt this schedule and to restart it asynchronously by the application. The TTCAN protocol allows for this feature. If the application in the current time master wants to interrupt the periodic schedule and the system designer has enabled this feature the time master signals in the next reference message that after the end of this basic cycle a gap has to expected. If so, the other nodes will not interpret the missing reference message after the end of the basic cycle as an error and in particular the potential backup time masters will not send a reference message. If the application of the time master wants to restart the periodic schedule again it initiates the sending of a reference message synchronized to the corresponding event.

6. Future system architectures and TTCAN

The need for deterministic message transfer within future, sometimes safety relevant systems, can be carefully planned and realized by the means of the TTCAN exclusive time windows. The sequence of different base cycles within a TTCAN system matrix allows multiple combinations of sending patterns, e.g. for x-by-wire messages with different timing periods. The service of a fault-tolerant global time and the implementation of a drift correction based on a time master principle support the requirement for system wide time base in distributed real-time applications. The necessary support of OSEKTime OS specification and OSEKTime FTCom specification is also guaranteed. TTCAN specification focuses on the extension of the existing CAN standard modules. The intelligent use of the already included mechanisms in CAN, like the bitwise arbitration, allows to bring more flexibility within the deterministic framework. TTCAN allows the definition of arbitrating time windows for sending spontaneous messages. The deterministic start of the next periodic message is guaranteed by the protocol specification and supervised by built in error mechanisms. TTCAN is best suited for system solutions which do not necessarily need redundant communication channels managed by the communication protocol and where the data rate and redundancy requirements do not exceed the CAN limitations, e.g. the first generation of xbws with mechanical/hydraulic backup.

7. Summary

This document describes the extension of CAN with the time triggered execution of the CAN protocol. Due to the time triggered approach the communication structure in TTCAN is deterministic and hence seems more suitable to fulfill requirements of future applications and system architectures in vehicles while still maintaining flexibility during development as well as in use. TTCAN also supports the needs of other industries, where the time triggered approach, determinism and global network time is required, e.g. medical services and automation industry. Furthermore, TTCAN still allows the use of the CAN based monitoring and analyzing tools. The technical knowledge of the engineers about CAN is still necessary and only has to be updated for the TTCAN extensions.

The current specification of TTCAN does not cover all requirements of safety related distributed systems (redundancy or data transmission rate), e.g. for dry xbws. Further development steps concerning these characteristics are checked in the context of the ISO TC22.

8. Literature

1. The Time-Triggered Approach to Real-Time System Design; H. Kopetz, TU-Wien;
2. TTP – A Protocol for Fault-Tolerant Real-Time Systems; H. Kopetz, G. Grünsteidl; IEEE Computer; January 1994, pp. 14-23.
3. OSEKTime: A dependable Real-Time Fault-Tolerant Operating System and Communication Layer as an enabling Technology for By-Wire Applications; St. Poledna, M. Glück, Ch. Tanzer, TTTech; S. Boutin, Renault; E. Dilger, Th. Führer, Robert Bosch GmbH; Ch. Ebner, BMW Technik; E. Fuchs, DeComSys; R. Belschner, B. Hedenetz, DaimlerChrysler; B. Holzmann, A. Schedl, BMW AG; R. Nossal, B. Pfaffeneder, Siemens AG; Th. Ringler, TU-Stuttgart; Y. Domaratsky, A. Krüger, Motorola; A. Zahir, ETAS GmbH; SAE 2000, Detroit, Michigan, U.S.A.
4. The Steer-By-Wire prototype implementation: Realizing time triggered system design, fail silence behavior and active replication with fault-tolerance support; Th.Führer, Robert Bosch GmbH; A. Schedl, DaimlerChrysler; SAE 1999, Detroit, Michigan, U.S.A.
5. Composability in the Time-Triggered Architecture; H. Kopetz, TU-Wien; SAE 2000, Detroit, Michigan, U.S.A.
6. Road vehicles – Interchange of digital information – Part 1: Controller area network data link layer and medium access control; ISO 11898.
7. Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit; ISO 11898.
8. Road vehicles – Controller area network (CAN) – Part 4: Time triggered communication; Working Draft ISO 11898.
9. Guaranteeing Message Latencies on Controller Area Network (CAN); K. Tindell, A. Burns; Proceedings 1st International CAN Conference; 1994; pp 2-11.
10. Real-Time Systems: Design Principles for Distributed Embedded Applications; H. Kopetz; Kluwer Academic Publishers; 1997; ISBN 0-7923-9894-7.

Dipl.-Ing. Thomas Führer
Robert Bosch GmbH
P.O.Box 10 60 50
70049 Stuttgart
Germany
Phone: +49 711 811-7597
Fax: +49 711 811-7136
Email: Thomas-Peter.Fuehrer@de.bosch.com

Dr. Bernd Müller
Robert Bosch GmbH
P.O.Box 10 60 50
70049 Stuttgart
Germany
Phone: +49 711 811-7053
Fax: +49 711 811-7136
Email: mueller.bernd@de.bosch.com