

## **Time- and Priority-triggered Send/Receive Node**

Execution examples of the present new development concern a send/receive node, and in particular a send/receive node that can be coupled to a communication medium via an interface, and that provides an available message to the medium based on a time schedule and a send priority.

Computer communication in the area of real-time systems that are linked by wired data buses is usually based on event-triggered and time-triggered protocols. With event-triggered protocols, messages are sent when a triggering event occurs, e.g., if a switch is pressed or a threshold is exceeded. With time-triggered protocols, messages are sent at prescheduled points in time and thus can provide deterministic network behaviour. In case of sporadic traffic, event-triggered protocols use less network resources. However, only limited real-time guarantees are possible in case of a set of events occurring (almost) simultaneously, possibly at different nodes. In order to observe hard real-time guarantees, in case of periodic data traffic, time-triggered protocols are preferable. However, the use of time-triggered protocols for sporadic data traffic and small maximum response times results in massive waste of bandwidth, because worst cases have to be considered, e.g. maximum response time or the minimum time interval between events.

In a typical scenario in a car, networked components of chassis, power train, safety, comfort, and entertainment compete for network resources. Due to very different requirements with regard to bandwidth, delay, jitter, reliability, and real-time guarantees, several communication technologies are currently used together in the automotive domain. Among these technologies are CAN (Controller Area Network), TTCAN (Time-Triggered CAN), FlexRay, MOST (Media-Oriented System Transport), and LIN (Local Interconnect Network).

Figure 9 shows a known networking of real-time systems of a vehicle, using a variety of wired data buses. The networked components of the power train, e.g. engine control 10, gear box control 12, and vehicle dynamics 14, are usually linked via a high-speed CAN bus 16. The FlexRay bus 18 is used for by-wire systems for steering and braking and therefore connects e.g. vehicle dynamics 14, steering control 20, pedal node control 22, wheel node control 24, and servo control 26. Furthermore, entertainment and telematics components, e.g. radio 28, navigation device 30, phone 32, display 34, monitor 36, and parking assistant 38 are linked via a MOST bus 40. Networked components regarding comfort, safety, and simple chassis functionalities, e.g. fan

control 40, light control 42, seat control 44, air conditioning control 46, windshield wiper 48, door control 50, and control panel 52 are usually connected via a low-speed CAN bus 54. The LIN buses 56 and 58 can e.g. be used for the networking of further air conditioning and seat control components. In addition, the high-speed CAN bus 16, the MOST bus 40, and the low-speed CAN bus are connected to a control unit and gateway node 60, where the control unit 60 is further linked with the driver displays 62.

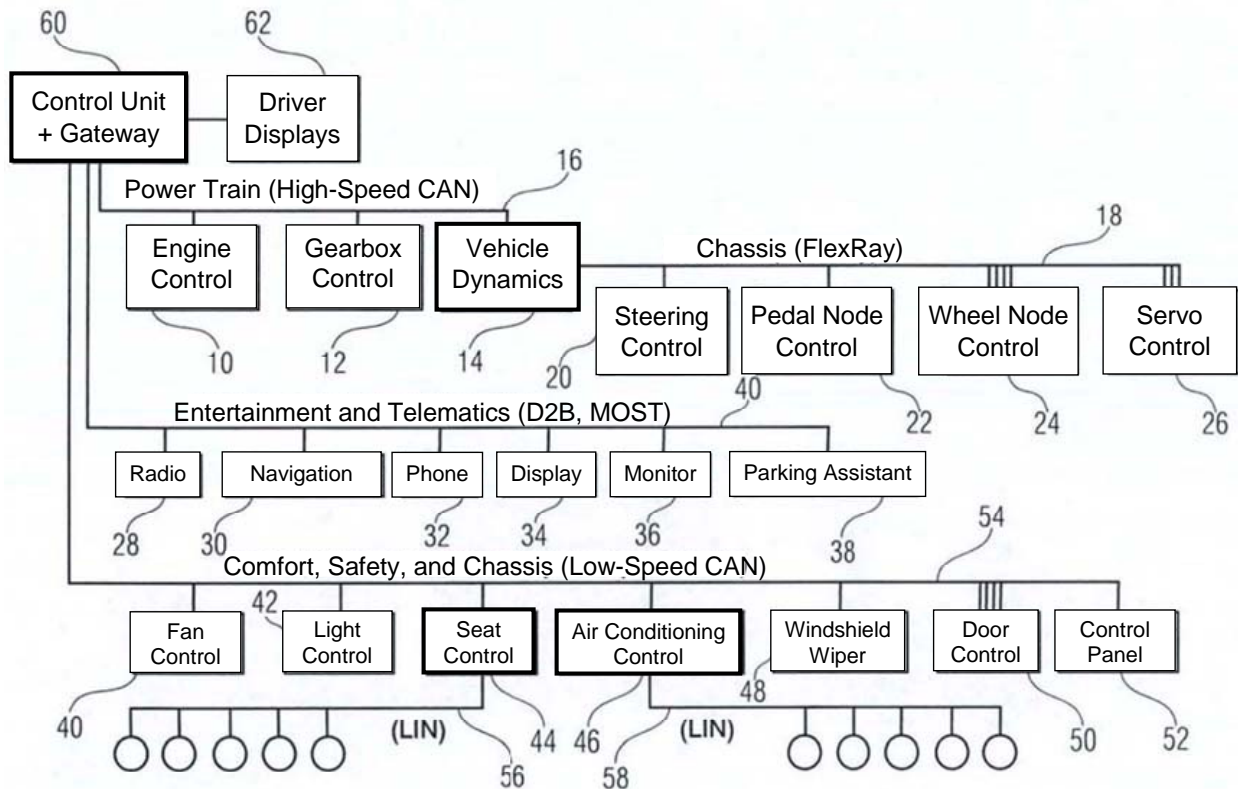


Figure 9

Moreover, cars are produced in many different configurations according to customer demands. These configurations may exhibit different communication patterns, depending on dynamic vehicle usage and road situations. To avoid extensive testing of all possible configurations in all possible scenarios, the communication system must support composability in the sense that communication properties are always maintained. For instance, adding further airbags, wheel sensors, or a sunroof must not inhibit the timely operation of other, safety-relevant components. A known solution of this problem is the temporal encapsulation of nodes, where fixed time slots are assigned to individual nodes prior to deployment. A drawback of this solution, however, is that communication systems have to be oversized, to cover all possible configurations and communication patterns.

To guarantee predictable message delays and communication bandwidths, precise scheduling and assignment of communication resources is done at development time, i.e. offline. In addition, to achieve deterministic real-time guarantees, worst-case situations must be considered. This results in a massive waste of resources in all normal situations, which usually require far less communication bandwidth.

Several methods for the improved capacity utilization of wired data buses are known in the automotive domain. For instance, the publication *SMART-Plan: A New Message Scheduler for Real-time Control Networks, ISSC 2005* by M. Naughton and D. Heffernan and the publication *Systematic Message Schedule Construction for Time-Triggered CAN, IEEE Transactions on Vehicular Technology, Special Issue on Vehicular Communication Networks, Vol. 56, Issue 6, Part 1, pp. 3431-3441, 2007* by K. Schmidt and E. G. Schmidt provide approaches for the computation of optimal time schedules for TTCAN. Furthermore, the publication *Message Scheduling for the FlexRay Protocol: The Dynamic Segment, IEEE Transactions on Vehicular Technology, Vol. 58, No. 5, pp. 2160-2169, 2009* by E. G. Schmidt and K. Schmidt, and the publication *Message Scheduling for the FlexRay Protocol: The Static Segment, IEEE Transactions on Vehicular Technology, Vol. 58, No. 5, pp. 2170-2179, 2009* by K. Schmidt and E. G. Schmidt provide approaches for the computation of time schedules for the dynamic and static segment of FlexRay, respectively. These approaches assume that access to a time slot is either exclusive (one node) or priority-based (all nodes), where given real-time requirements are to be satisfied.

Thus, the known methods to determined improved time schedules for event-triggered and time-triggered protocols are based on narrow restrictions – medium access is either exclusive (one node) or contention-based (all nodes). This substantially reduces the search space for improved solutions. Another drawback of known solutions is the priority-based medium contention, where all nodes are allowed to participate. This significantly reduces possible real-time guarantees in case of several (almost) simultaneous events.

Further known methods are dynamic assignment schemes where nodes have to communicate with some resource manager. This, however, causes a substantial amount of overhead, adds to the overall delays, and yields an uncertain outcome.

The present new development therefore has the objective to provide a send/receive node for a bus system with multiple access operating under real-time constraints, which supports real-time medium access with high reliability and relatively small resource requirements.

This objective is achieved by to a send/receive node according to C1, a bus system with a multitude of send/receive nodes according to C15, a method for the operation of a send/receive node according to C19, the use of a multitude of send/receive nodes according to C20, or a computer program according to C21.

The new development creates a send/receive node, which is devised to provide the frame of an available message with an assigned send priority to a communication medium via an interface, based on a time schedule with corresponding time slots, if in the corresponding time slot, there is no further available message of another send/receive node attached to the communication medium that has a higher send priority.

Execution examples of the new development describe a send/receive node that is devised to provide or signal the send priority of an available message on the communication medium. Furthermore, there are execution examples where the send/receive node is devised to receive another send priority of another available message of another send/receive node that is attached to the communication medium, and to compare the received send priority with the send priority of its own available message. In addition, the send/receive node is devised to provide the frame of the available message to the communication via the interface, if for the corresponding time slot, the send priority of its own available message is the highest priority. In execution examples, the interface of the send/receive node is further devised to communicate over a data bus with multiple access in real-time operation. Furthermore, execution examples of the new development describe a bus system with one data bus and a multitude of send/receive nodes.

Execution examples of the present new development will be explained subsequently, referring to the attached figures:

Figure 1a shows a schematic view of an execution example of a send/receive node with an interface to link to a communication medium;

- Figure 1b shows a schematic view of a send/receive node with an optional memory to store a time schedule with the assigned time slots and/or the send priority of the message to be provided;
- Figure 2 shows a schematic view of an execution example of a time slot associated with the send/receive node;
- Figure 3 shows a bus system with one data bus and a multitude of send/receive nodes;
- Figure 4 shows a schematic view of a structuring of the communication medium into time intervals;
- Figure 5 shows an execution example of a time slot assignment for a simplified scenario of a vehicle, in tabular form;
- Figure 6 shows an execution example of a frame assignment that is consistent with the time slot assignment, in tabular form;
- Figure 7 shows the structuring of the communication medium into time intervals for TTCAN;
- Figure 8 shows the structuring of the communication medium into time intervals for FlexRay;
- Figure 9 shows a known networking of real-time systems of a vehicle using a multitude of wired data buses.

In the following description of execution examples of the new development, identical or apparently identical elements of the figures are referred to using the same symbols, such that their description in the different execution examples is exchangeable.

Figure 1a shows a schematic view of an execution example of a send/receive node 100 with an interface 102 for the attachment to a communication medium. The send/receive node 100 further shows a processing direction 104, which is devised to record, for a sequence of time slots, the occurrence of a first and a second condition, and, if the first and the second condition are satisfied for a corresponding time slot, to provide a frame of an available message to the communication

medium via the interface 102 during the corresponding time slot, where for the transfer of the message, one or several frames (e.g. data frames) may be needed, and where a frame may have one or more logical values (bits). The number of needed frames per message and the number of needed logical values per frame may depend, for instance, on the size or the type of the message. The first condition specifies the assigned time slot for the provision of the frame of the available message, according to a time schedule, while the second condition specifies that the send priority associated with the message for the assigned time slot is higher than another send priority of another available message of another send/receive node that is attached to the communication medium.

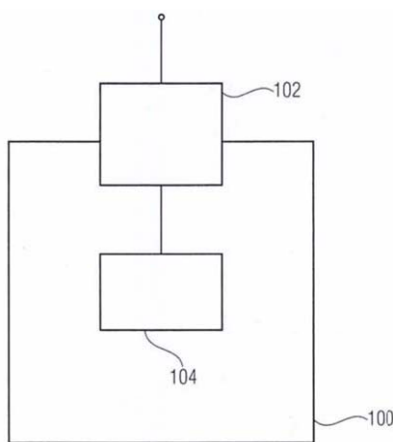


Figure 1a

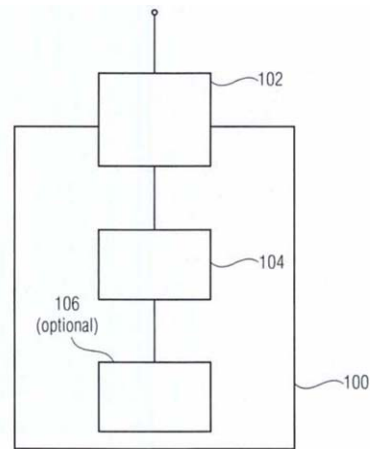


Figure 1b

Figure 1b shows a schematic view of a send/receive node 100 with an optional memory 106 to store a time schedule with the assigned time slots and/or send priority of the message to be provided. In execution examples, the memory 106 can, in addition, be devised such that the time schedule with the assigned time slots and/or send priority of the message to be provided, stored in the memory 106, can be updated. Furthermore, the time schedule may have one or more time slots that are assigned to the send/receive node 100. Based on the time schedule with the assigned time slots and based on the send priority of the message to be provided, the processing unit 104 may record when and whether the frame of the available message is to be provided to the communication medium via the interface 102.

The send/receive node 100 may, in execution examples, be further devised to provide the send priority of the available message or an information derived thereof to the communication medium via the interface 102. Furthermore, the send/receive node 100 may be devised to receive the send priority or an information derived thereof of another message of another send/receive node

attached to the communication medium via the interface 102, and to compare the received send priority or an information derived thereof with the send priority of its own available message or an information derived thereof. The information derived from the send priority may, for instance, incorporate an encoded version of the send priority, or a value associated with the send priority, where for instance higher values may mark higher send priorities, or vice versa. Subsequently, the notion of send priority will be used for better understandability, where in execution examples, the send priority or an information derived from the send priority may be used.

If the received send priority of the assigned time slot has a lower preference than the send priority of the own available message, then the assigned time slot may be used by the send/receive node 100 to send or provide the frame of the available message, respectively. Furthermore, the send/receive node 100 may also receive several send priorities, and compare the received send priorities with the send priority of its own available message. Thus, the send/receive node that has the message with the highest send priority wins this limited competition for the assigned time slot and may transmit the frame of the available message.

Furthermore, the send/receive node 100 may be devised, in execution examples, to provide no send priority to the communication medium via the interface 102, if the send/receive node 100 has no available message for the assigned time slot. Thus, the send/receive node 100 does not participate in the limited competition for the assigned time slot, such that another send/receive node attached to the communication medium that has a message with, for instance a lower preference or send priority may use the assigned time slot for the transmission of the frame of the available message with lower preference. This enables a significantly more effective utilization of the available bandwidth of the communication medium, while observing real-time guarantees at the same time.

Furthermore, the send/receive node 100 may be devised, if the first condition is satisfied and the send/receive node 100 has several available messages with different send priorities, to determine the available message with the highest send priority, and to provide the determined highest send priority to the communication medium via the interface 102. Thus, if a send/receive node 100 has several messages with different send priorities, there may at first be a local competition among the messages of the send/receive node 100. Here, local competition means that the send/receive node 100 evaluates the send priorities of its own available message to, for instance, determine the highest send priority. The winner of the local competition is, for instance, the message with the

highest send priority. Afterwards, the send/receive node 100 provides the highest send priority determined for the assigned time slot to the communication medium via the interface 102, whereby a global competition arises, if another or several other send/receive nodes have an available message for the assigned time slot. Here, global competition means that the send/receive node 100 receives and evaluates the send priority/priorities of the other available messages of the other send/receive nodes attached to the communication medium. The winner of this global competition is the message with the highest send priority.

In the following execution examples, it is assumed that the send/receive node 100 always signals the send priority of the message that has won the local competition. Furthermore, unless stated otherwise, the notion of competition always refers to the global competition, where several send/receive nodes may participate.

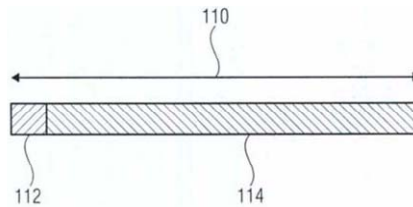


Figure 2

Figure 2 show a schematic view of an execution example of a time slot 110 assigned to a send/receive node 100. The time slot 110 assigned to the send/receive node 100 may be decomposed in subsection, where the send/receive node 100 is devised to provide, in a first subsection 112, to provide the send priority of the available message to the communication medium via the interface 102, and, in another subsection 114, to provide the frame of the available message via the interface 102, if no higher send priority is received. Therefore, the send/receive node may, for instance, be devised to perform the comparison of the send priority of its own message with the received send priorities during the first subsection of the assigned time slot. However, the new development is not limited to such an execution example; rather, the time slot 110 may, in addition, be subdivided into further subsections of possibly different length.

In case the send/receive node 100 has an available message, but loses the limited competition for the assigned time slot, because a higher send priority of another message of another send/receive node 100 that is attached to the communication medium is received, the send/receive node 100 may be devised to store the available message in order to provide the available message to the

communication medium via the interface 102 in a subsequent assigned time slot. This may be necessary, if the information to be transmitted continues to be relevant, or if another send/receive node that is attached to the communication medium relies on this information. Alternatively, the send/receive node 100 may be devised to discard the available message in this case. This may be the case if the information to be transmitted is out of date at a later point in time, or no more needed. Whether an available message is stored or discarded, may, for instance, be determined at development time (e.g., of the vehicle), and usually depends on the information contained in the message. For example, the message of a temperature sensor containing a temperature value may usually be discarded, if a message with an updated temperature value is available for the next assigned time slot. A message for the adjustment of a seat or mirror position may, for instance, still be relevant, wherefore such a message may, for instance, be buffered and transmitted in a subsequent assigned time slot.

Furthermore, the send/receive node 100 may be devised, in execution examples, to synchronize the time schedule of the assigned time slot and/or the send priority of the message to be provided with a central control unit, which has the time schedule and the send priorities of the messages to be provided, in a synchronization time slot. The updating of the time schedule with the assigned time slot and/or the send priority may, for instance, occur when the send/receive node or the data bus are powered up, for instance, after restart of the vehicle. Furthermore, the send/receive node may be devised to perform the update stand-alone or upon request. Furthermore, the send/receive node 100 may have a clock or pulse generator and/or means or devices for tick or pulse synchronization with further send/receive nodes that are attached to the communication medium.

In execution examples, the interface 102 of the send/receive node 100 may further be devised to establish a connection with the data bus. Therefore, in the following, execution examples of the send/receive node 100 with a data bus and a plurality of send/receive nodes 100 are described.

Figure 3 shows a bus system 120 with a data bus 122 and a plurality of send/receive nodes  $100_1$  until  $100_n$ . The bus system 120 extends, for given real-time requirements, the search space for the discovery of optimal time schedules for time slotted data buses with multiple access. Therefore, the concept of execution mode (mode for short) is introduced. Execution modes first of all define the preference of system functionalities, and may thereby also define the send priority of messages of the corresponding functionality. In a bus system, there usually are messages of higher relevance and messages of lower relevance, for instances, message concerning safety, or messages

containing an audio stream. Therefore, for instance, the execution modes *emergency*, *safety*, *regular*, and *stream* may be introduced. Execution modes can be sorted by preference, and define, for instance, priorities of messages that are associated with these execution modes. In addition, an optimal, static, mode-based slot schedule may be determined offline, such that the required schedulability is granted, and at runtime (e.g. during the time a frame needs from sender to receiver) no additional data are to be transmitted, i.e. there is no additional overhead. For instance, in the execution example shown in Figure 5, four modes are used. However, the new development is not limited to such execution examples; rather, an arbitrary number of modes may be used.

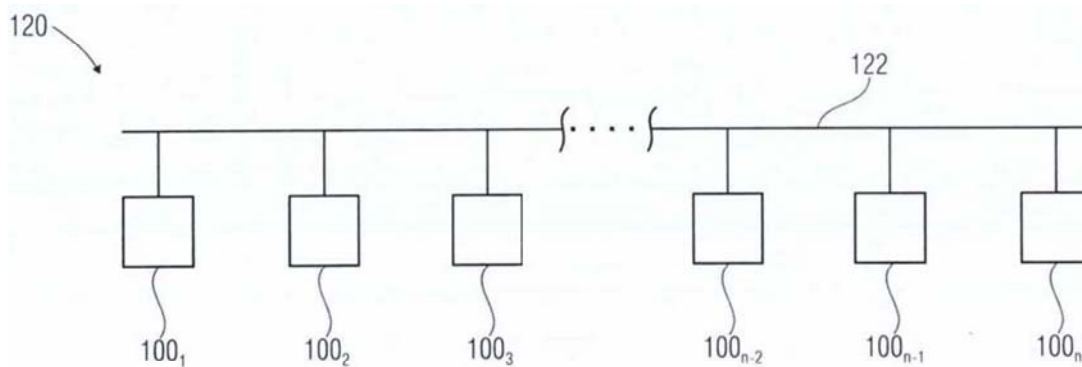


Figure 3

The introduction of execution modes opens up, on the one hand, potential for the improvement of the system structuring; on the other hand, the execution modes form the basis for mode-based slot scheduling. The mode-based slot schedule determines, for each time slot, a number of messages that are potentially to send or to be provided, where at most one message is scheduled per execution mode and time slot. Thereby, there is limited contention in each time slot, sorted by priorities, which ensures a flexible usage of assigned time slots at runtime. The bus system with a data bus and a plurality of send/receive nodes 100 thus enables a limited controlled degree of dynamic contention in each time slot. With mode-based slot scheduling, a time slot may be assigned to several send/receive nodes 100 for diverse intended uses. Because each occurrence of these time slots can, for instance, only be used once, it must be ensured that the intended uses are not in conflict. A preferred approach can be to identify execution modes that are mutually exclusive, or that can be sorted by preferences, and to assign the time slot at most once per execution mode. Messages of higher relevance, e.g. safety-relevant messages, and messages of lower relevance, e.g. an audio data stream, may be sorted by preference and be assigned to an execution mode. For instance, it can be distinguished between an execution mode *stream* and an execution mode *safety*, whereupon safety-relevant messages in the execution mode *safety* get

preference over entertainment messages in execution mode *stream*. Different from the scheduling approaches currently used in the automotive domain, mode-based slot scheduling enables controlled multiple assignments of the same time slot. Thereby, the available time slots are better utilized, which leads to an improved utilization of communication resources.

Thus, mode-based slot scheduling is a global scheduling approach for time slots with multiple assignments in distributed communication environments. Below, an abstract communication model is defined, which may be instantiated by a variety of already existing time-triggered communication protocols, such as, for instance, FlexRay and TTCAN, where the following assumption (Def. 1) is made: A single-hop network consists of a finite set  $V = \{v_1, \dots, v_s\}$  of send receive nodes (nodes)  $100_1$  until  $100_n$ , which are pair wise connected by links.

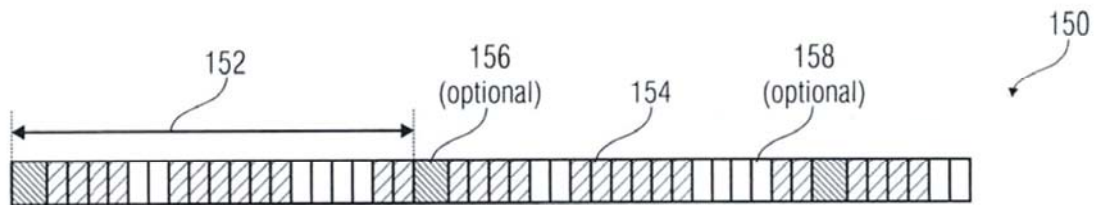


Figure 4

In this context, Figure 4 shows a schematic view of a time structuring of the communication medium 150 into time intervals. The mode-based slot scheduling is based on a communication medium (medium) 150, which can be decomposed into a sequence of time intervals (TDMA, time division multiple access). In execution examples, it is assumed that the communication medium 150 is decomposed into subsequently numbered macro slots 152 of some length, for instance, the same length, whereupon macro slots 152 are in turn decomposed into numbered micro slots 154. In the following, macro slots 154 are called time slots 154, where time slots 154 may correspond to the time slots 110 in the previous execution examples 1 to 3.

A macro slot 152 may, in addition, correspond to, for instance, a FlexRay communication cycle, or a basic cycle of TTCAN. Assignments refer to micro slot numbers and are valid for the occurrence of the corresponding time slot 154 in each macro slot 152. Thus, the following assumption (Def. 2) can be made: Time is decomposed into an infinite sequence  $S^* = (S_1, S_2, \dots)$  of some length, called macro slots 152. Each macro slot  $S$  and accordingly 152 is in turn subdivided into a finite sequence  $S = (s_1, \dots, s_n)$  of micro slots 154 that may have different length.

Each micro slot 154 has one occurrence per macro slot 152. The occurrence of micro slot  $s_j$  (154) in macro slot  $S_i$  (152) is referred to as  $s_{i,j}$ .

The structuring of the communication medium (medium structuring) 150 underlying mode-based slot scheduling can be seen as a virtual structuring, omitting irrelevant detail that may have some importance otherwise. For instance, a macro slot 152 may additionally include an interval for time synchronization and idle micro slots 158. In the execution example shown in Figure 4, only the areas shaded grey represent micro slot occurrences that are assigned for transmission. In Figure 4, as an example,  $n=12$  micro slots 154 are assigned for transmission.

In the following, transmission modes are introduced. Transmission modes correspond to execution modes and determine the preference and accordingly the priority of messages, whereupon in a bus system, messages may be weighted according to their relevance. Furthermore, transmission modes may, for instance, be derived from corresponding system execution modes. Modes are assigned a unique mode preference, which determines the priority of the corresponding messages and accordingly data streams. Here, the following assumption (Def. 3) can be made: Transmission modes are modeled as a non-empty, finite set  $M = \{m_1, \dots, m_r\}$ . To each mode, a unique mode preference  $mp: M \rightarrow \mathbf{N}_0$  is assigned, where lower values may represent higher preferences. To mark the preference, other conventions may be determined, too.

For each mode, micro slots 154 may be assigned to send/receive nodes  $100_1$  until  $100_n$ , which may use these micro slots 154, for instance, to send frames of available messages or provide them to the communication medium 150 via the interface 102. This means that given  $r$  modes  $m_k$ , a micro slot 154 can be assigned up to  $r$  times, possibly to different send/receive nodes  $100_1$  until  $100_n$ . If micro slot  $s_j$  of mode  $m_k$  is assigned to some send/receive node  $v$  (100), for instance, to send/receive node  $100_2$ , this implicitly applies to the occurrences of this micro slot 154 in all macro slots 152. Thus, the following assumption (Def. 4) can be made: Let  $V$  be a single-hop network,  $M$  be a set of modes, and  $S$  be a sequence of micro slots 154. Then a partial function, which defines the assignment of micro slots ( $SA = \text{slot assignment}$ ) can be defined as follows:

$$SA: S \times M \rightarrow_p V$$

The assignment of micro slots ( $SA$ ) can, e.g. be determined statically and defines all bandwidth allocations. If  $SA$  is defined for some pair  $(s,m)$ , this uniquely defines a send/receive node 100, for

instance, the send/receive node  $100_2$ , to which all occurrences of slot  $s$  ( $154$ ) are assigned in mode  $m$ . If  $SA$  is undefined for some pair  $(s,m)$ , the occurrences of slot  $s$  ( $154$ ) are not used in mode  $m$ . Thus, the definition of slot assignment ( $SA$ ) precisely states the first condition, which defines the point in time at which the frame of the available message is provided.

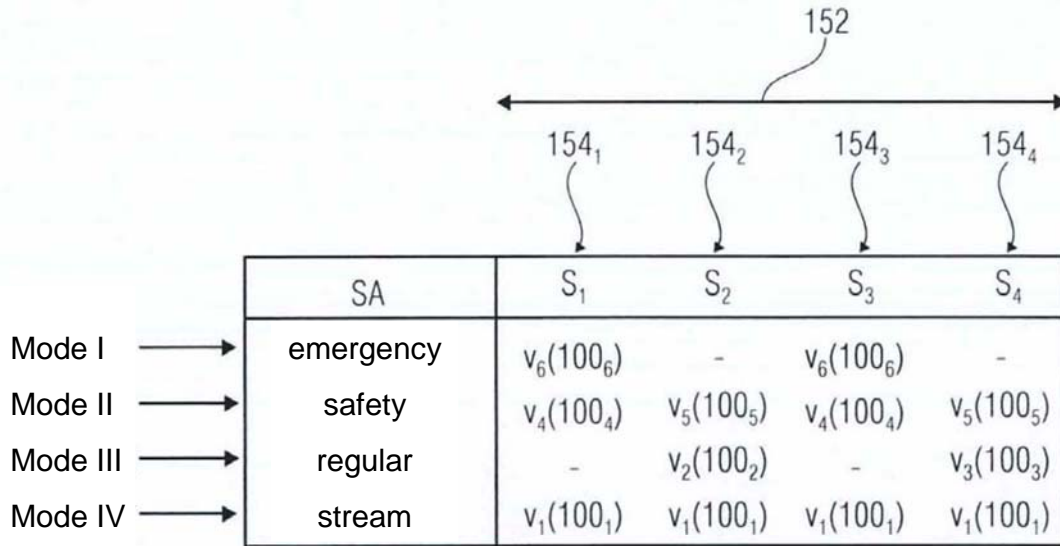


Figure 5

Figure 5 shows an execution example of a slot assignment in tabular form for a simplified scenario of a vehicle. The table has four micro slots  $154_1$  until  $154_4$  per macro slot  $152$  and four modes. Mode I: *emergency* with mode preference one; Mode II: *safety* with mode preference two; Mode III: *regular* with mode preference three, and Mode IV: *stream* with mode preference four. According to the following basic principle, the micro slots  $154_1$  until  $154_4$  are assigned to six different send/receive nodes  $100_1$  until  $100_6$ , where for illustration purposes, it is assumed that each send/receive node  $100_1$  until  $100_6$  has only messages of one message type. However, the new development is not constrained to such execution examples; rather, if a send/receive node has messages of different preferences, there first of all may be a local competition.

In the execution example shown in Figure 5, a send/receive node  $v_1(100_1)$  periodically sends entertainment messages, for instance, an audio data stream, which may completely use the available bandwidth. Therefore, all micro slots  $154_1$  until  $154_4$  are assigned to the send/receive node  $v_1(100_1)$  in Mode IV *stream*. The send/receive nodes  $v_2(100_2)$  and  $v_3(100_3)$  send messages with sporadic events, which for instance indicate a mirror movement or a seat positioning, in Mode III *regular*, which has preference over Mode IV *stream*. Since these events are rare, the micro slots  $154_2$  and  $154_4$  are usually not used by the send/receive nodes  $v_2(100_2)$  and  $v_3(100_3)$ .

The send/receive nodes  $v_4$  ( $100_4$ ) and  $v_5$  ( $100_5$ ) send messages that indicate steer-by-wire and brake-by-wire events, respectively, in Mode II *safety* with high preference. The send/receive node  $v_6$  ( $100_6$ ) signals events of an airbag in Mode I *emergency*. Since these events are extremely rare, they may event get preference over events and therefore messages of the steer-by-wire system. Here, the following assumptions (Def. 5) can be made: Let  $F$  be a set of frames of available messages, i.e. messages to be sent,  $S^*$  be a sequence of macro slots 152,  $S$  be a sequence of micro slots 154,  $M$  be a set of modes, and  $V$  be a single-hop network. Then a partial function, which defines an assignment of frames ( $FA =$  frame assignment) can be defined as follows:

$$FA: S^* \times S \times M \times V \rightarrow_p F$$

Frame assignment ( $FA$ ) determines which frames are scheduled for which slot 154, mode, and send/receive node 100. In practice, this will be decided dynamically, as the frames of available messages are not known in a state where the send/receive nodes  $100_1$  until  $100_n$  are not ready (offline). With mode-based slot scheduling, a frame of an event-triggered message is scheduled only, if the corresponding event occurs and the send/receive 100 therefore has an available message. For instance, messages caused by events of the brake-by-wire system are scheduled only if the driver uses the brake pedal. Also, frames of available messages may have to be scheduled more than once, if they compete with other frames of available messages associated with a mode of higher preference. In other words, if a frame of an available message is scheduled for a given micro slot 154, this does not imply that it will actually be sent in that slot 154.

In execution examples, the assignment of a frame must be consistent with a slot assignment, i.e. frame assignment must respect slot assignment. Here, the following assumption (Def. 6) can be made: Let  $SA$  be a slot assignment,  $FA$  be a frame assignment.  $FA$  is consistent with  $SA$  iff

$$\forall S \in S^*, s \in S, m \in M, v \in V, f \in F. ( FA(S,s,m,v) = f \Rightarrow SA(s,m) = v )$$

Here,  $S$  is a sequence of micro slots 154,  $S^*$  is a sequence of macro slots 152,  $M$  is a set of modes,  $m$  is a mode,  $V$  is a single-hop network,  $v$  is a send/receive node 100,  $F$  is a set of frames of available messages, and  $f$  is a frame of an available message. Thus, the strategy of mode-based signaling precisely states the second condition, which defines that the send priority associated with the frame of an available message scheduled for the assigned time slot is higher than another

send priority of another frame of another available message of another send/receive node attached to the communication medium.

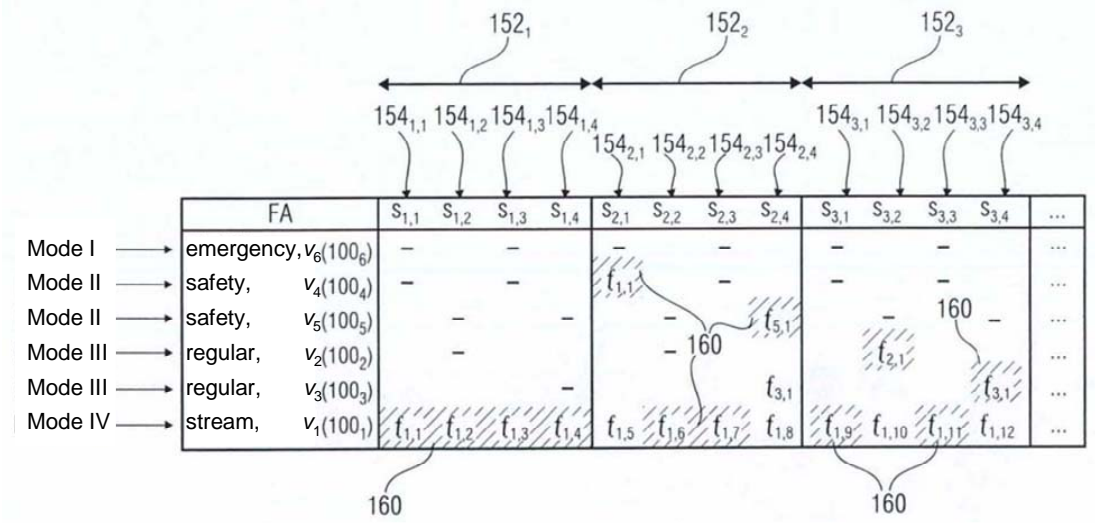


Figure 6

Figure 6 shows an execution example of a frame assignment that is consistent with the slot assignment in Figure 5, in tabular form. Scheduled frames of available messages are represented by an entry  $f_{i,j}$ , with the first index corresponding to the index of the send/receive node  $100_i$  until  $100_6$ , and the second index being a consecutive numbering. In Figure 6, macro slots 152 have identifiers with consecutively indexes, while identifiers of micro slots 154 have two indexes, with the first index referring to the corresponding macro slot 152, and the second index being a consecutive numbering within the macro slot 152. Assignments of micro slots 154 without a scheduled frame are represented by an entry “-“. In the execution example shown in Figure 6, the send/receive node  $v_1$  ( $100_1$ ) is ready to use all available bandwidth, whereas the other send/receive nodes  $100_2$  until  $100_6$  have only sporadic needs.

Thus, the new development concept of mode-based scheduling can be defined as follows (Def. 7): Let  $FA: S^* \times S \times M \times V \rightarrow F$  be a frame assignment,  $tx \subseteq S^* \times S \times V \times F$  be a relation defining whether a frame is sent by a node in a given slot occurrence 154 by a send/receive node 100. Then mode-based scheduling is the strategy where, for each slot occurrence 154, the scheduled frame of the available message with highest mode preference is sent. Formally:

$$\forall S \in S^*, s \in S, m \in M, v \in V. \exists f \in F. (f = FA(S, s, m, v) \wedge \forall m' \in M, v' \in V, f' \in F. (m' \neq m \wedge f' = FA(S, s, m', v') \wedge mp(m') < mp(m)) \Rightarrow tx(S, s, v, f))$$

Here,  $S$  is a sequence of micro slots 154,  $S^*$  is a sequence of macro slots 152,  $s$  is a micro slot 154,  $M$  is a set of modes,  $m$  and  $m'$  are modes,  $V$  is a single-hop network,  $v$  and  $v'$  are send/receive nodes 100,  $F$  is a set of frames of available messages,  $f$  and  $f'$  are frames of available messages,  $FA$  is a frame assignment,  $mp$  is the mode preference function, and  $tx$  denotes whether a send/receive node 100 sends the frame of an available message during the assigned time slot 154.

From the above definitions, it follows that in each occurrence of a micro slot 154, at most one send/receive node 100 can be sending. Competition for an assigned micro slot 154 is always won by the send/receive node 100, which has the message with the highest send priority and accordingly the mode with the highest preference. For each time slot, at most one message per mode is scheduled, such that in a corresponding time slot, it is never the case that two or more send/receive nodes 100 have available messages with the same send priority. In the execution example shown in Figure 6, for instance, the send/receive node  $100_4$  can send in micro slot  $154_{2,4}$ . In addition, the shaded areas 160 in Figure 6 indicate the frames that are sent according to the mode-based schedule. Therefore, with mode-based scheduling, one time-triggered communication medium is sufficient for the slot assignment shown in Figure 6, whereupon for the realization of a completely deterministic schedule, more than one communication is required, depending on real-time requirements.

Whether a frame of an available message is sent or not, is a distributed decision, whereupon all send/receive nodes  $100_1$  until  $100_n$  are devised such that they decide unanimously, based on their local information. If, for an assigned micro slot 154, several messages compete, the message associated with the mode that has highest preference wins. Frames of available messages losing the competition are either rescheduled or discarded; however, these are application-specific decisions not addressed by mode-based scheduling. In the execution example shown in Figure 6, the frames of available messages in Mode IV *stream*, e.g. frames  $f_{1,5}$  and  $f_{1,8}$ , are discarded, while frames of available messages in Mode III *regular*, e.g. frame  $f_{3,1}$ , are rescheduled. As long as no messages are available in Mode I *emergency*, frames of available messages in Mode II *safety* win the competition, thus, no rescheduling is necessary in these cases. Since it is required for each time slot 154 that the frame is sent which has the message with the mode with highest priority, the fast signaling of the current mode is a basic precondition for the efficient implementation of mode-based scheduling.

In execution examples, the current mode with the highest preference is propagated fast and reliably across the network and accordingly the data bus, to efficiently implement mode-based scheduling as described above. In the following, this is called fast mode-signaling. In addition, it can be ensured by fast mode-signaling, that the current transmission mode is known to all send/receive nodes  $100_1$  until  $100_n$  at the beginning of each time slot or with a very short (negligible) delay. Fast mode-signaling can be achieved in different ways, depending on the scope of transmission modes.

In an execution example according to the new development, with a bus system consisting of one data bus and a plurality of send/receive nodes  $100_1$  until  $100_n$  using mode-based scheduling, the scope of transmission modes is limited to individual send/receive nodes 100. This leads to a controlled competition among send/receive nodes 100, whereupon each send/receive node 100 can determine the frame assignment for the assigned time slot 154. So, if there is a non-empty set of scheduled frames of available messages, then the mode with the highest preference is the local mode that is to be applied to the assigned time slot 154. To send the frame of the available message, the send/receive node 100 competes with other send/receive nodes  $100_1$  until  $100_n$ , by signaling the mode and accordingly the send priority of the available message in a collision-protected way. Thus, fast mode-signaling enables bus-wide deterministic signaling of the current mode with the highest priority right at the beginning of a time slot and therefore supports real-time communication. This concept of fast mode-signaling is especially suited for the efficient implementation of mode-based scheduling.

The new development concept of fast mode-signaling has significant advantages compared to the system-wide or global scope of transmission modes. With global scope, only time slots associated with the current transmission mode can be used. Furthermore, it would be necessary to signal the global mode, for instance, at the beginning of each macro slot, or at the beginning of each micro slot. Thereby, frames of available messages that are associated with the current mode could be sent immediately and without any further competition. However, if no available message is associated with the current mode, the time slot can not be used as long as the current mode persists. Moreover, transmission modes with global scope are in conflict with mode-based scheduling, since an efficient usage of the available bandwidth is no longer ensured.

In the following, the new development concept of fast mode-signaling is described by two execution examples for the implementation of mode-based scheduling in TTCAN, and FlexRay.

Since the new development concept is highly efficient, no additional control messages are required.

The first execution example shows an efficient implementation of mode-based scheduling in TTCAN, where, for a better understanding, the basic functionalities of CAN and TTCAN are outlined first. For medium arbitration, the CAN bus applies a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) scheme, which uses dominant and recessive bits. MAC frame transmission (MAC = Medium Access Control) starts with a dominant SOF-bit (Start-Of-Frame bit), followed by an 11-bit CAN message identifier determining the frame priority, which together form the arbitration sequence. During the arbitration sequence, a node compares the bit detected on the medium with the transmitted bit. If a recessive bit has been sent, but a dominant bit is observed, meaning that another node is sending an identifier with higher priority, the node backs off. Thus, if several nodes start a MAC frame transmission (almost) simultaneously, the node sending the frame of highest priority will always win the arbitration, provided CAN message identifiers are unique. However, it is possible that frames with high priority defer low-priority frames indefinitely. The TTCAN is based on the CAN bus, however, different from CAN, exhibits deterministic behavior.

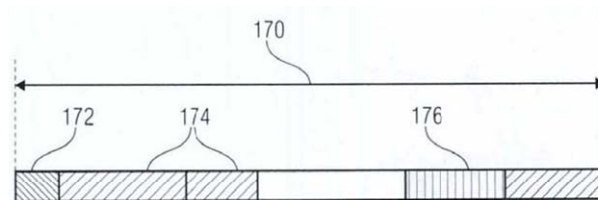


Figure 7

Figure 7 shows the time structuring of the communication medium of TTCAN. With TTCAN, time is divided into basic cycles 170, where a basic cycle 170 corresponds to a macro slot 152. Relative to the reception times of a beacon frame (called reference message 172) at the beginning of each macro slot, time slots (called time windows) that may have different length are defined. Time windows are *exclusive*, *arbitrating*, or *free*. Exclusive time windows 174 can be assigned to one predefined message of a particular node. If the system is properly configured, no contention with other nodes will occur. For arbitrating time windows 176, the regular CAN arbitration scheme is applied, which can be used for event-triggered, spontaneous messages.

TTCAN with mode-based scheduling, as shown by the present new development, can be realized by implementing fast mode-signaling based on the available mechanisms of TTCAN. Fast mode-signaling can e.g. be achieved by applying regular CAN arbitration in exclusive time windows 174, by associating, with each mode, a CAN message identifier in such a way that mode preferences are respected. To realize mode-based scheduling for the slot assignment in Figure 1, for instance, it is sufficient to use e.g. four CAN message identifiers. Thus, whenever several send/receive nodes 100 attempt to send frames in the same assigned exclusive time window 154, the frame of the available message with highest mode preference will always win the contention. Contention in exclusive time windows 174 is feasible, because according to the CAN standard, regular CAN arbitration remains active.

In comparison to regular CAN operation, the operation of TTCAN with mode-based schedules has two crucial differences. First, contention in each exclusive time window 174 is limited to a well-defined, controllable number of send/receive nodes 100 – up to one send/receive node 100 per mode. If an exclusive time window 174 is assigned to one send/receive node 100 and one mode only, the time window usage is exclusive, and operation is fully deterministic. Second, CAN message identifiers that encode modes can be used for different message types sent by different send/receive nodes 100, as long as messages of these types are sent in different time windows. This would violate regular CAN operation, but is not harmful under mode-based scheduling, if each mode is used at most once in each time window, a property that can easily be checked at configuration time.

With TTCAN and mode-based scheduling, it is usually not necessary to exploit the full reservoir of CAN message identifiers. If  $n_{modes}$  denotes the number of modes, the length of the arbitration phase can be reduced to  $\lceil \log_2 n_{modes} \rceil$  bits. For instance, in case of 4 modes, 2 arbitration bits are sufficient, in case of 15 modes, 4 arbitration bits are needed. This optimization can, for instance, be realized through a modification of the TTCAN standard.

To illustrate the differences between TTCAN with mode-base scheduling and pure TTCAN, the execution example shown in Figure 5 can be used. If mode-based scheduling is implemented as described above, a solution with only one TTCAN bus is feasible. In case only exclusive time windows 174 are used, a total of three TTCAN buses are required, under the assumption that independent of the actual set of payload data, entire time slots are used.

Furthermore, TTCAN with mode-based scheduling avoids the following disadvantages as compared to a combined solution using pure TTCAN. In a combined solution, using only one TTCAN bus, exclusive time windows 174 could for instance be assigned to periodic data of Mode IV *stream*, while arbitrating time windows 176 are used for sporadic data of Modes I to III *emergency*, *safety*, and *regular*, with suitable preferences implemented by assigning CAN identifiers. To obtain some bandwidth for these sporadic data, we can assign one time window per basic cycle, thereby reducing bandwidth for periodic data. This combined solution creates the following problems: Emergency messages compete with safety messages, which in turn compete with regular messages in the same time windows, and therefore could delay them indefinitely. Moreover, it is required that different message types of the same mode, but originating from different send/receive nodes 100, e.g. by-wire-systems such as, e.g. steer-by-wire or brake-by-wire, use unique CAN message identifiers, one for each message type. This, however, could result in situations where messages of one type, e.g. steer-by-wire messages, delay messages of another type, e.g. brake-by-wire messages. Especially in by-wire-systems, this is not acceptable.

As another execution example, the new development concept of fast mode-signaling is exemplarily described for the implementation of mode-based scheduling in FlexRay, in particular for a protocol that is derived from FlexRay or an enhancement of FlexRay, which is called FlexRay in the following. For a better understanding of the implementation, the basic functionality of FlexRay is described first.

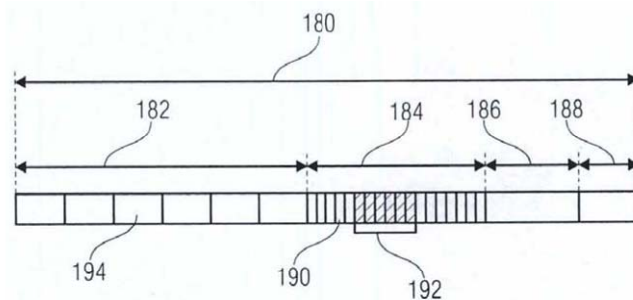


Figure 8

Figure 8 shows the time structuring of the communication medium of FlexRay. Time is divided into macro slots called communication cycles 180. A communication cycle 180 is in turn divided into four statically configurable segments, a static segment 182, a dynamic segment 184, an optional symbol window 186, and the Network Idle Time (NIT) 188. The static segment 182 has a number of static time slots that can be assigned to network nodes for deterministic access. The

dynamic segment 184 is subdivided into mini slots 190, which can be associated with frames by their number. Transmission of a frame with identifier  $k$  starts in the  $k^{\text{th}}$  free mini slot 190 of the dynamic segment 184, if the remaining part of the dynamic segment 184 is sufficiently long to complete the transmission. For the transmission itself, several mini slots 190 are assembled into a dynamic time slot 192; during this period, counting of free mini slots 190 is suspended. This scheme gives preference to frames with lower identifiers, but does not guarantee that all sporadic frames competing in a dynamic segment 184 can be sent. The optional symbol window 186 is used for network management purposes. Symbols in FlexRay are encoded as low pulse phases of defined length. To protect them against distortions, only the length of the low pulse phase is evaluated. Together with (recessive) idle phases, symbols could be used for binary encoding. The mandatory *Network Idle Time (NIT)* 188 is used for correction of clock offset and related internal computations. In addition, several basic cycles 180 of FlexRay can be combined into a matrix cycle.

FlexRay with mode-based scheduling, as shown by the present new development, can be realized by implementing fast mode-signaling based on the available mechanisms of FlexRay. For the implementation, time slots 194 of the static segment 182 can, for instance, be decomposed into mini slots, where each mini slot is associated with a transmission mode such that mode preferences are respected. To send the frame of an available message, the send/receive node 100 listens on the communication medium from the beginning of the static time slot 194. If the communication medium is idle until the mini slot that is associated with the transmission mode of the available message, the send/receive node 100 has won the competition and can send the frame of the available message.

An advantage of this approach is that, for instance, existing FlexRay adapters can be used to implement fast mode-signaling. However, the solution tends to become inefficient with a growing number of modes, as each mode consumes one mini slot in each static time slot 194. To improve the situation, the number of modes can, for instance, be limited. If the number of modes can not be limited, then, for instance, for each time slot 194, a slot assignment with a small number of modes could be used. Therefore, a transmission mode such that mode preferences are respected is associated, with each mini slot, on a per-static-slot basis. This means that the same mode may be associated with different mini slots in different static time slots 194. Alternatively, mode preferences can be binary-coded, and be transmitted as a sequence of (dominant) FlexRay symbols and (recessive) idle phases at the beginning of static slots 194. This introduces active

contention among send/receive nodes 100 willing to use a static time slot 194, similar to the scheme applied for CAN arbitration.

To illustrate the differences between FlexRay with mode-base scheduling and pure FlexRay, the execution example shown in Figure 5 can be used. If mode-based scheduling is implemented as described above, a solution with only one FlexRay bus is feasible. In case only static time slots 194 are used, a total of three FlexRay buses are required, under the assumption that FlexRay arbitrates only one bus.

Furthermore, FlexRay with mode-based scheduling avoids the following disadvantages as compared to a combined solution using one FlexRay bus. In a combined solution, using only one FlexRay bus, static time slots 194 could for instance be assigned to periodic data of Mode IV *stream*, while dynamic time slots are used for sporadic data of Modes I to III *emergency*, *safety*, and *regular*, with suitable preferences. However, the combined solution requires that the communication cycle is extended by a dynamic segment 184, which reduces the available bandwidth for periodic messages. Moreover, the combined solution has the property that sporadic messages with high preference can delay other event-triggered messages with lower preference indefinitely.

For the adherence of real-time guarantees in the execution example shown in Figure 5, a solution that is based on a FlexRay bus or a TTCAN bus with mode-based scheduling and fast mode-signaling is therefore feasible, but there is no solution that is based on a pure FlexRay bus or a pure TTCAN bus. Mode-based scheduling with fast mode-signaling therefore enables a significantly more efficient utilization of the available bandwidth of TDMA networks in real-time scenarios. This is of particular interest in the automotive domain, where a tendency towards time-triggered protocols is noticeable. The bus system with one data bus and a plurality of send/receive nodes 100 can thus be used to network the plurality of send/receive nodes in the form of sensors, controllers, input and/or output devices in vehicles.

In execution examples, mode-based schedules are realized by fast mode-signaling on a wired, time-triggered data bus with multiple access. Mode priorities are realized during medium arbitration such that after the (very short) arbitration phase, only one send/receive node 100 is sending. In this way, for a given application profile, real-time guarantees can be granted as far as this is required. Furthermore, execution examples enable an improved bandwidth utilization of

wired data buses in real-time operation. The new development concept of execution modes, mode-base scheduling and fast mode-signaling thus enables a solution for wired, time-triggered data buses with multiple access with real-time capabilities. To this end, a static time schedule, controlled competition per time slot and very fast dynamic signaling of the current mode with highest preference are used.

Moreover, the execution examples of the new development are fully compatible with current time scheduling technologies of the automotive industry, which only enable purely exclusive or purely arbitrated access to the communication medium. Furthermore, current time scheduling technologies of the automotive industry can be covered as special cases of mode-based scheduling, e.g. by the exclusive assignment of a time slot to a particular send/receive node 100. For this reason, the existing communication systems can be used as starting point for the implementation of mode-based scheduling, by realizing a controlled competition in a suitable subset of exclusive time slots. The new development concept thus leads to a significantly higher bandwidth utilization of bus systems, without loss of real-time capabilities. Thereby, depending on the application area, it is possible, for instance to reduce the number of necessary bus systems and/or to get by with less bus systems. With the help of the execution examples, it has been shown that the prerequisites for the realization of the present new development are already given; therefore, mode-based signaling is not coupled to technological preconditions. In addition, mode-based signaling is compliant with currently used scheduling approaches.

Execution examples of the new development have been described where the send/receive node 100 is devised to provide the send priority of the available message to the communication medium in the assigned time slot, and to receive another send priority of another message of another send/receive node that is attached to the communication medium. Moreover, the send/receive node 100 which has the message with the highest send priority can, for instance, be devised to signal the send priority in the assigned time slot, but also to send the frame of the available message immediately, without receiving another send priority and comparing it with its own send priority. This is possible, because the send/receive node 100 that has the message with highest priority always wins the competition for the assigned time slot. Furthermore, the send/receive node that has the message with the lowest send priority may, for instance, be devised to provide the frame of the available message to the communication medium via the interface in the assigned time slot, if no further send priority can be received, without signaling the send priority of the

available message. This is possible because the send/receive node 100 that has the message with the lowest send priority always loses the competition for the assigned time slot.

Furthermore, execution examples have been described where the new development concept of fast mode-signaling has been exemplarily used for the implementation of mode-based scheduling for TTCAN and FlexRay. However, the new development is not limited to such execution examples; rather, fast mode-signaling can be used for the implementation of mode-based signaling, for example, for CAN, LIN, MOST, and D2B, for communication technologies defined by the standards ISO 10681, ISO 11898, ISO 11992, ISO 11783, SAE J2284, and SAE J1939, and for other technologies.

Moreover, execution examples of the new development have been described where the new development concept of fast mode-signaling has been used for the implementation of mode-based scheduling for communication technologies of vehicles. However, the new development is not limited to such execution examples; rather, the new development concept of fast mode-signaling can also be used for the implementation of mode-based scheduling for communication technologies and corresponding protocols and standards of aircrafts, e.g. for the implementation of mode-base scheduling for TTP (Time-Triggered Protocol), ARINC 629, and ARINC 825 (ARINC = Aeronautical Radio Incorporated).

Furthermore, execution examples have been described where a plurality of send/receive nodes 100 are connected in a single-hop network. However, the new development is not limited to such execution examples; rather, several bus systems with one data bus and a plurality of send/receive nodes 100 may be combined into a multi-hop network, for instance, via gateway nodes, where higher level addressing schemes and routing protocols are required. To ensure deterministic behavior of multi-hop networks, time slots of the different bus systems may, for instance, be arranged such that messages that are received by gateway nodes can be relayed within a short and predictable amount of time over the next hop. The principle to determine suitable mode-based schedules for multi-hop networks is the same as for mode-based signaling in single-hop networks. The new development concept of fast mode-signaling is also applicable to wired multi-hop networks.

Protective Scope

- C1. Send/receive node (100) with an interface (102) to establish a connection with a communication medium, where the send/receive node (100) is devised to determine, for a sequence of time slots, the occurrence of a first condition and a second condition, and, if the first and the second condition are satisfied for an assigned time slot, to provide a frame of an available message via the interface (102), where the first condition determines the assigned time slot for the provision of the frame of the available message according to a time schedule, and where the second condition determines that the send priority associated with the messages for the assigned time slot is higher than another send priority of another available message of another send/receive node attached to the communication medium.
- C2. Send/receive node (100) according to C1, where the send/receive node (100) is devised to provide the send priority of the available message or an information derived thereof to the communication medium via the interface (102), if the first condition is satisfied and the send/receive node (100) has an available message.
- C3. Send/receive node (100) according to C1 or C2, where the send/receive node (100) is devised to determine the available message with the highest send priority, and to provide the determined highest send priority or an information derived thereof to the communication medium via the interface (102), if the first condition is satisfied and the send/receive node (100) has several available messages with different send priorities.
- C4. Send/receive node according to one of C1 to C3, where the send/receive node (100) is devised to receive another send priority or an information derived thereof via the interface (102), and to compare the received send priority or information derived thereof with the send priority of its own available message or an information derived thereof in order to determine whether the second condition is satisfied.
- C5. Send/receive node (100) according to one of C1 to C4, where the second condition is also satisfied, if the send/receive node (100) has not received another send priority from the communication medium via the interface (102) in the assigned time slot.

- C6. Send/receive node (100) according to one of C1 to C5, where the assigned time slot (110) of the send/receive node (100) is divided into subsections (112), where the send/receive node (100) is devised to provide, in a first subsection (112), the send priority of the available message or the information derived thereof to the communication medium via the interface (102), and to provide, in another subsection (114), the frame of the available message to the communication medium via the interface (102).
- C7. Send/receive node (100) according to C6, where the send/receive node (100) is devised to perform the comparison during the first subsection (112) of the assigned time slot (110).
- C8. Send/receive node (100) according to one of C1 to C7, where the send/receive node (100) is devised to store available message, if only the first condition is satisfied, in order to provide the available message to the communication medium in a subsequent assigned time slot.
- C9. Send/receive node (100) according to one of C1 to C7, where the send/receive node (100) is devised to discard the available message, if only the first condition is satisfied.
- C10. Send/receive node (100) according to one of C1 to C9, where the send/receive node (100) has a memory (104) to store the time schedule with the assigned time slots or the send priority of the messages to be provided.
- C11. Send/receive node (100) according to C10, where the time schedule with the assigned time slots or the send priorities stored in the memory (104) can be updated.
- C12. Send/receive node (100) according to C10 or C11, which is devised to synchronize, during a synchronization time slot, the time schedule with the assigned time slot or the send priorities with a central control unit that has the time schedule and the send priorities of messages.
- C13. Send/receive node (100) according to one of the C1 to C12, which has a clock or pulse generator and/or means or devices for tick or pulse synchronization with further send/receive nodes.
- C14. Send/receive node (100) according to one of C1 to C13, where the interface (102) is devised to communicate with a data bus with multiple access in real-time operation.

- C15. Bus system (120) with a data bus (122) and a plurality of send/receive nodes (100) according to one the C1 to C14.
- C16. Bus system (120) according to C15, where the data bus (122) is devised as a data bus with multiple access for real-time operation.
- C17. Bus system (120) according to C16, where the data bus (122) is devised to connect a plurality of send/receive nodes (100) in the form of sensors, control devices, input and/or output devices in vehicles or aircrafts.
- C18. Bus system (120) according to one of C15 to C17, where the occurrence of the first and second condition can be represented in matrix form, where the first dimension of the matrix has the time slots of the time schedule for providing the frames, and the second dimension of the matrix has the provided send priorities of the messages, and where the provided send priorities occurring the second dimension of the matrix are different.
- C19. Method for the operation of a send/receive node, which has an interface to establish a connection with a communication medium, with the following features:
- Determining of the occurrence of a first condition and a second condition for an assigned time slot of a sequence of time slots; and
- Providing of a frame of an available messages to the communication medium during the assigned time slot via the interface, if the first and the second condition for the assigned time slot are satisfied,
- where the first condition determines the assigned time slot for the provision of the frame of the available message according to a time schedule, and where the second condition determines that the send priority associated with the messages for the assigned time slot is higher than another send priority of another available message of another send/receive node attached to the communication medium.
- C20. Use of a plurality of send/receive nodes (100) according to one of C1 to C14 in a bus system with a data bus with multiple access for real-time operation, where the data bus is devised as a protocol according to CAN, TTCAN, FlexRay, LIN, MOST, D2B or TTP, or as a standard

according to ISO 10681, ISO 11898, ISO 11992, ISO 11783, SAE J2284, SAE J1939, ARINC 629 or ARINC 825.

C21. Computer program for the execution of the method according to C19, if the computer program runs on a computer or a micro processor.

### Summary

The present new development creates a send/receive node with an interface to establish a connection with a communication medium, where the send/receive node is devised to determine, for a sequence of time slots, the occurrence of a first condition and a second condition, and, if the first and the second condition are satisfied for an assigned time slot, to provide a frame of an available message via the interface to the communication medium during the assigned time slot, where the first condition determines the assigned time slot for the provision of the frame of the available message according to a time schedule, and where the second condition determines that the send priority associated with the messages for the assigned time slot is higher than another send priority of another available message of another send/receive node attached to the communication medium.

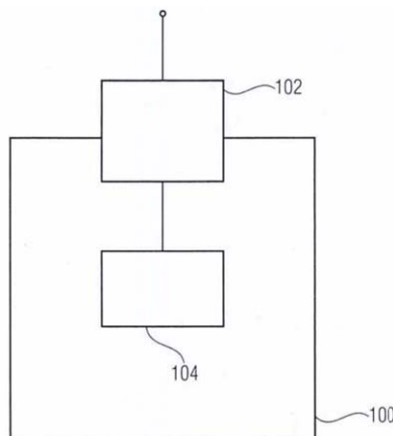


Figure 1