Fault tolerant TTCAN networks

B. Müller, T. Führer, F. Hartwich, R. Hugel, H. Weiler, Robert Bosch GmbH

TTCAN is a time triggered layer using the CAN protocol to communicate in a time triggered fashion. As TTCAN is based on CAN it uses the power of CAN´s error detection mechanisms and robustness, but it also provides a step towards determinism and time triggered technology.

Future system architectures will include applications that need to access more than one TTCAN controller. This article describes how to build fault-tolerant TTCAN networks, in particular the mechanisms to synchronize different TTCAN busses. It is shown that it is very easy to implement a synchronized network of any reasonable redundancy level, even if non-trivial architectures (for instance more than a simple dual channel network) are involved. Moreover, this synchronization can be achieved even when the individual TTCAN busses use different time bases without ever violating the modular integrity of one single bus.

1 Introduction

There is a variety of real-time bus-systems that are used to connect electronic control units in automation or in the automobile. Most of these communication protocols are one channel systems, i.e. although there are possibly some fault-tolerance mechanisms, there is no really redundant transmission of messages. In some safety critical applications however, redundant message transmission becomes a requirement.

A time triggered variant of CAN, denoted in the sequel by TTCAN, is described by the ISO standard 11898-4 (currently still a draft version). Essentially CAN and hence TTCAN is a one channel system, redundancy can only be provided by using multiple TTCAN busses. However, compared with intrinsically redundant systems (e.g. FlexRay, TTP/C), the use of multiple single channel busses introduces the problem of management of redundancy. This mainly consists of synchronizing the different busses, but it must also be ensured that the main services of a time triggered communication system (providing a global time and a consistent schedule all over the network) can be used by an application from either of the channels. This means it must be possible for an application to treat the set of different busses as one communication system. In the paper it is shown that the TTCAN interfaces allow to easily combine TTCAN busses in a modular way so that this can be achieved even in system architectures that go far beyond the standard dual channel scenario.

2 TTCAN

As fault tolerant TTCAN networks consist of combinations of TTCAN busses we begin with a short description of the TTCAN bus. The interested reader may find more detailed descriptions in [1].

2.1 TTCAN Basics

Time triggered communication in TTCAN is based on the reference message being transmitted regularly by the time master. Following the reference message there is a sequence of time windows that provide the time slots for individual message transmissions. There are three types of time windows: exclusive time windows that are exclusively reserved for one message, arbitrating time windows during which messages can compete for the bus by the non-destructive arbitrating mechanism of CAN, and free time windows that are reserved for future extensions of the network. The pattern of time windows following a reference message is called a basic cycle, i.e. each basic cycle starts with a reference message and contains an offline configured set of time windows.

In TTCAN not all basic cycles necessarily have to be the same. It is possible to distinguish different basic cycles by the cycle count, a counter that is incremented each cycle up to the maximum value after which it is restarted again. Combining all these different cycles we get the so called matrix cycle which represents the complete communication overview of a TTCAN network.

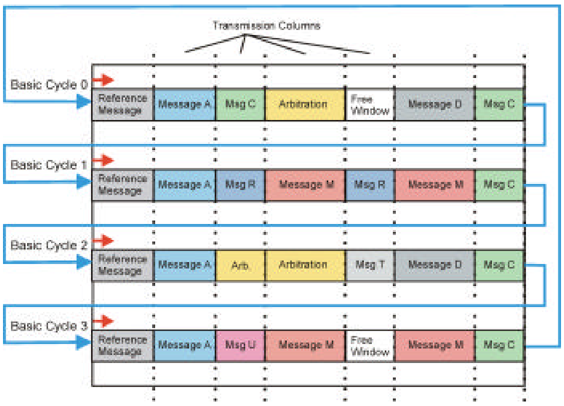

Figure 1 Example of a matrix cycle

2.2 TTCAN Timing

A TTCAN controller can be configured to support two levels, level 1 and level 2. Level 2 is an extension of level 1, and only in level 2 high end synchronization and global time are provided [1]. In the sequel we will always consider level 2 networks although some of the mechanisms described below can also be used in level 1 networks.

Within a TTCAN controller we basically have three time notions: local time, cycle time, and global time. All three times run with the same rate but use different and varying relative phases to each other. Local time is the controller internal basis for all other times.

The basic network time unit is the so called NTU and within a level 2 controller the local time counter must be able to count even fractional parts of an NTU (with a fractional resolution of at least 3 bits).

With each reference message, cycle time is restarted. More precisely: The sample point of the SoF bit of any message generates a (logical) frame synchronization pulse in the network. On occurrence of such a frame synchronization pulse each controller captures the current value of its local time and, after identification of a message as reference message, treats this value (the local reference mark) as the starting point of cycle time, i.e. within a controller we have

Cycle time = Local time – Local reference mark.

Actually the fractional parts of this difference are ignored as cycle time only counts NTUs.

Byte 1: Next_Is_Gap bit and cycle count

7	6	5	4	3	2	1	0
Next_Is_ Gap	Reserved	Cycle Count (5)	Cycle Count (4)	Cycle Count (3)	Cycle Count (2)	Cycle Count (1)	Cycle Count (0)

Byte 2: Fractional part NTU_Res of Master_Ref_Mark and discontinuity bit

7	6	5	4	3	2	1	0
NTU_Res (6)	NTU_Res (5)	NTU_Res (4)	NTU_Res (3)	NTU_Res (2)	NTU_Res (1)	NTU_Res (0)	Disc_Bit

Byte 3: Low byte of Master_Ref_Mark

7	6	5	4	3	2	1	0
MRM (7)	MRM (6)	MRM (5)	MRM (4)	MRM (3)	MRM (2)	MRM (1)	MRM (0)

Byte 4: High byte of Master_Ref_Mark

7	6	5	4	3	2	1	0
MRM (15)	MRM (14)	MRM (13)	MRM (12)	MRM (11)	MRM (10)	MRM (9)	MRM (8)

Figure 2: Reference message in TTCAN level 2

Global time is not only using the existence but also the content of the reference message. Figure 2 shows the four mandatory bytes of the reference message, further data bytes may be added by the application.

By definition, at a given time, global time in a TTCAN network is what the current master thinks it is. So within the reference message the master transmits the global time value including fractional parts (the Master_Ref_Mark) that is valid at the pulse of the respective frame synchronization. The local offset - the difference between the Master_Ref_Mark and the corresponding local reference mark - gives the difference between local and global time at the point of frame synchronization, hence the local approximation to global time is given by

Global time = local time +local offset.

This even holds for the time master. We see that both cycle time and global time are derived from local time, and both are resynchronized with every reference message. Again the fractional parts of this difference are ignored as cycle time only counts NTUs.

Moreover, each controller adjusts by use of the clock synchronization algorithm the rate of its local time, so that it almost perfectly matches the rate of the master: Within a TTCAN controller the rate of local time is determined by the so called TUR (time unit ratio) variable, a variable that indicates the (non-integer) ratio between an NTU and the local clock period. TUR has a configuration value within each controller, but a perfect adjustment to an oscillator in another node can only occur if the TUR value is adapted so that the local length of an NTU equals the global (the master´s) length of the NTU as good as possible. The TTCAN clock synchronization automatically computes the optimum TUR value by use of the global time given by the master and therefore ensures that the different local times within the network have the same rate.

2.3 TTCAN Initialization

In a single channel TTCAN network there can be up to 8 potential time masters, distinguished by the three bit time master priority. The time master priority is given by the three least significant bits of the reference message that is transmitted by the respective potential time master. Although we do not describe the initialization procedure in detail (see for instance [1]), it is important to note that eventually the potential time master with the highest time master priority becomes the time master of the TTCAN network.

2.4 The Gap Case

Although it is possible to transmit the reference message completely periodically this is not required in TTCAN. To support this the time master announces in the reference message by use of the Next_Is_Gap bit that after the current basic cycle there will be a gap of undetermined length. (However, the maximum length of the gap is specified off-line). At some point the application initiates the transmission of a reference message and so brings the gap to an end. This typically is synchronized with some application specific event.

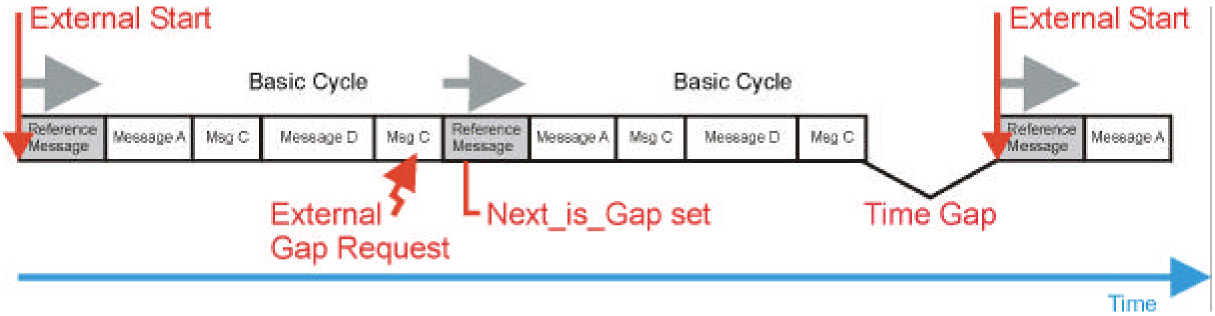

Figure 3: The gap

This feature is particularly useful if one wants to synchronize applications to processes that are not periodic in the TTCAN time. Note that the gap does not influence continuity of global time, i.e. global time just continues to increase during the gap as it does during a cycle.

2.5 Global time discontinuities

By definition the relation between TTCAN's global time and any external time is determined by the more or less random time at which the initialization of the TTCAN network began. In some applications it is desirable to synchronize TTCAN global time to some external source. In some cases the external source is not always present, in particular not during initialization (consider for instance synchronization to GPS time during start of an automobile in a garage). In this case at some point of time there will be a TTCAN network running with a global time that is significantly different from the external time and a synchronization problem arises. A TTCAN controller allows the application (of the master) to add some value to the global time, but, as the TTCAN global time is also used for protocol internal clock synchronization reasons, this must be signaled to the other nodes in the network. This signal is given by setting the discontinuity bit. The synchronization procedure contains the following steps:

- The application determines the difference between external time and TTCAN global time.
- The master is told to add this difference to the TTCAN global time.
- When transmitting the next reference message, the master uses the "new" global time within the reference message and sets the discontinuity bit.
- The receivers of the reference message adapt their global time according to the reference message but do not update their TUR value.

After these steps the TTCAN network is synchronized to the external global time.

2.6 Maintaining synchronization

The above procedure describes how to synchronize the TTCAN global time to an external source by a discontinuous jump. However, to maintain synchronization it is important to adjust the rates as well. As all nodes will follow the master, it is sufficient to adjust the rate of the master. This can be done by allowing the application to influence the TUR value of the master. By using both mechanisms it is very easy to build and maintain a high quality synchronization of the TTCAN global time to an external source.

3 Fault-tolerant TTCAN networks

Although CAN is intrinsically a two wire system with strong error detection and handling capabilities it is usually considered a one channel system. This means that fault-tolerance and redundant channels can only be achieved by combining two or more CAN busses. Naturally exactly the same is true for TTCAN. So this section deals with the problem how to combine two or more TTCAN busses. The subsequent definitions may seem extremely formal when considering the comparably obvious architectures they cover, however they allow a precise notation and they allow to demonstrate clearly the extremely strong fault-tolerance capabilities that can be achieved by a combination of TTCAN networks.

We call a system of two TTCAN busses a coupled TTCAN pair if there is at least one "gateway" node that has access to both TTCAN busses.

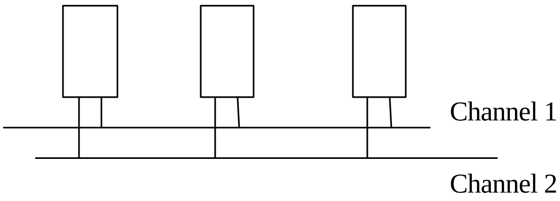

Figure 4: Classical redundant network

Figure 4 shows a typical redundant network, where each node is a "gateway" node. This is the most straight forward example of a coupled TTCAN pair. However, a coupled TTCAN pair covers also architectures as depicted in Figure 5.

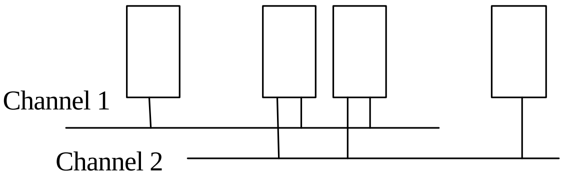

Figure 5: Mixed redundant network

Within a system of many TTCAN busses we call any two of those TTCAN-coupled if they are connected by a chain of coupled TTCAN pairs. More precisely: bus_a and bus_b are TTCAN coupled if there is a sequence $(bus_1,...,bus_n)$ of TTCAN busses with $bus_1 = bus_a$ and $bus_n = bus_b$ where (bus_i,bus_{i+1}) are coupled TTCAN pairs for all i=1,...,n-1. Now a fault tolerant TTCAN network is a system of TTCAN busses where each two of them are TTCAN coupled.

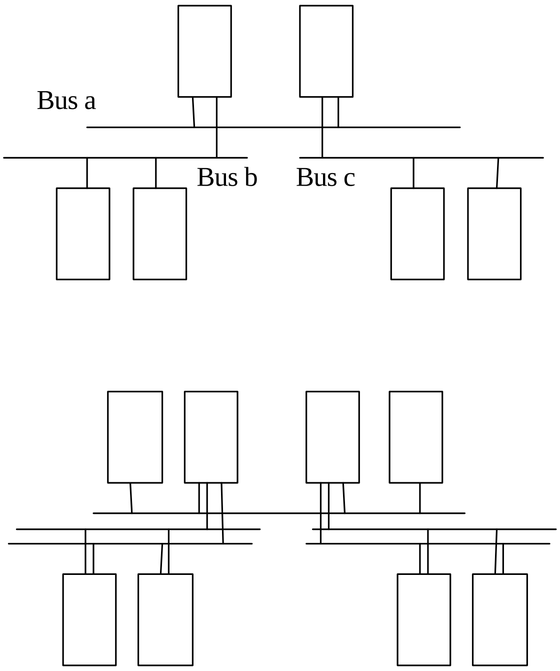

Figure 6: Examples for fault-tolerant TTCAN networks

Using fault-tolerant TTCAN networks it is possible to map the system safety structure into a system architecture that is best suited to solve the original problem.

The main basic message of the paper is that it is possible to treat all the busses of a fault-tolerant TTCAN network as one communication system.

4 Synchronization of TTCAN busses

The main problem of a combination of two time triggered busses is that of synchronization. Redundancy management on message level itself is not a communication system problem as even in a two channel system it should be possible to transmit different messages on the two busses, so the message memory is provided for each channel and the management of redundant transmission is the task of a higher layer (e.g. FTCom in OSEK-time). When synchronizing two TTCAN busses there is a variety of synchronization problems to solve. First of all each controller has three times. Local time essentially is a controller internal time so cycle time and global time remain as candidates for synchronizing. Secondly both of those times have a rate and a phase aspect to synchronize. Now by definition the rates of global and cycle time within a controller are equal as they are both derived from local time. So three synchronization problems remain

- Phase synchronization of cycle time
- Phase synchronization of global time
- Rate synchronization

As for one TTCAN bus all nodes will follow the time master in all three aspects it is sufficient to solve these problems for the master of the TTCAN bus. In the sequel for shortness we use the abbreviation SL for the synchronization layer that synchronizes the different TTCAN busses.

4.1 Phase synchronization of cycle time

After initialization each TTCAN bus is running with its own time master, cycle time, and global time. We first consider only two busses. First of all the SL has to measure the phase difference between the two busses. This can be done in any of the “gateway” nodes at any time as a phase (essentially) is not time dependent. To formulate this differently: As cycle time is part of the interface of a TTCAN controller the phase difference measurement is extremely simple for the SL and the real time requirements for this measurement are negligible for practical purposes.

A few remarks on the possible generalizations and implications should be added. First of all in order to have something like a “phase” we implicitly might assume that the (nominal) cycle lengths on the two busses are equal. Although this certainly will be one of the most important applications, this is not at all necessary. Consider for instance the case where the cycle length of the first bus is twice the cycle length of the second. It is obvious that a phase concept still is reasonably well defined and that the measurement still is as easy as in the case when all cycle lengths are equal. One might even go for more complicated scenarios like two cycles on the one bus versus three on the other.

The second point is that we also may have assumed that the time unit, the NTU, is the same on both busses. Again this certainly will be an important application case but it is not a necessary assumption. For instance in some applications the resolution for one bus can be much higher than on the other. This even extends to the use of different baud rates. As long as the SL knows the ratio between the involved NTUs there is still no problem. When thinking about a standardization of the SL one doubtlessly will restrict the possibilities but the fact that the definition of the TTCAN interfaces removes all non trivial real-time requirements from the measurement remains true.

After having performed the measurement the SL has to tell the time masters of the two busses the desired phase jump for the relevant bus. All of the above statements about real-time implications hold here as well. The desired phase difference might be zero, but this is not necessary. On the contrary in a redundant system one might explicitly wish to have a non zero phase difference to avoid common mode failures.

We make some remarks on possible implementations and special cases. In the easiest and most obvious case one bus synchronizes on the other, i.e. the first bus is not influenced at all by the SL and only the second bus adjusts its phase. If the current time master of the second bus is one of the gateway nodes, the SL can operate solely within this node and the necessary information transfer is more or less trivial. So an efficient implementation of the SL might use the following strategy: Bus$_2$ synchronizes on bus$_1$ (considers bus$_1$ as the "master bus"), the time master(s) of bus$_2$ with the highest time master priorities are gateway nodes, the SL operates within each of those gateway nodes internally (without any communication to the node outside), and the SL within a node can only get active when it detects that the node is the current master of bus$_2$. However, this strategy is the only one. It is possible to use CAN messages to communicate the desired phase jump to the current master of bus$_2$. In particular it is not necessary that one of the gateway nodes is a time master on any of the participating busses.

A time master that has to produce a phase shift will set the Next_Is_Gap bit in the next reference message and will use the desired phase jump to initiate the transmission of the reference message one cycle later. After the transmission of this deferred reference message on both busses the cycle times of both busses have the desired phase relation. Again by the interfaces of the TTCAN controller [2] this can be handled completely without any significant real-time implications on the involved application CPU. So the problem of synchronizing two TTCAN busses can be considered to be solved. When considering a whole fault-tolerant TTCAN network, the general case can by definition be reduced to the above, although the SL then additionally has to ensure that two TTCAN busses that are synchronized at some time stay so during the rest of the synchronization process. Without additional network load this can for instance be solved by introducing some kind of ordering structure (not even necessarily linear) on the system of TTCAN busses. Just to give an example: Each TTCAN bus is assigned an order level. There is one bus with order level 1, and each bus of order level n must form a coupled TTCAN pair with some bus of order level n-1. By definition of a fault-tolerant TTCAN network it is clear that such an ordering structure can always be found. The synchronizing strategy then is that each bus of order level n considers exactly one bus of order level n-1 as a master bus. This certainly is just an example but existence of one example shows that it is possible to handle the cycle time synchronization within a complete fault-tolerant TTCAN network.

4.2 Phase synchronization of global time

Essentially similar statements and strategies can be applied as for the cycle time although naturally another mechanism for the synchronization must be used. Therefore this part is only sketched. As above it is sufficient to consider only two busses. The SL has to measure the phase difference, calculate the desired phase shifts for both busses and give the information to the current time master of the respective bus. Both busses can have different cycle lengths and bit times as those are irrelevant for global time. Different nominal NTU lengths can easily be handled if there is an integer factor (preferably a power of 2) between them. The TTCAN interfaces allow to do this any time without imposing any real-time requirements onto the application CPU.

A time master that has to produce a phase shift uses the discontinuity bit for the global time exactly as explained in section 2.5. After successful transmission of this reference message the global time phases of the two busses are synchronized.

4.3 Rate synchronization

The rate both of global and cycle time on a TTCAN bus is determined by the rate of the local time of the current time master on this bus. Besides oscillator frequency the only relevant influence on the rate of the local time is given by the TUR value. Within the time master the TUR value remains constant during normal operation of the protocol (slow changes towards the configuration value are allowed, but this is not relevant here). However the application is allowed to enforce some TUR value using a TTCAN interface. This again allows a very simple way to synchronize rates (as above we can concentrate on the synchronization of two busses): The SL starts to measure the rate difference (or ratio) between both busses, it then calculates the new TUR values (or a corresponding quantity) for the two busses and gives this information to the current time masters of the respective busses. A time master receives a new TUR value from the application and, beginning with the transmission of the next reference message, starts to use this value. One basic cycle later the other nodes on the bus are rate synchronized as well.

Some comments on the measurement step: This can be done in different ways. A precise measurement can for instance be performed by measuring the length of the same physical time interval in units of both busses. Using interrupts (this has real-time implications, i.e. may not be a preferred option) one can use a time interrupt from one TTCAN controller at a given time to capture the local time of the other TTCAN controller. Doing this twice, we get the desired ratio. Without interrupts, the SL may capture the time values of both controllers on a regular (but not necessarily periodic) basis and get the ratio out the differences. There are a few variations of this strategy as well (like monitoring the behavior of time differences). Taking all things together it is possible to measure the ratio without any real-time implications. However, it may be possible that a division is required (acceptable in software). An optimized strategy can be used again if the second bus is synchronized on the first one and if the SL operates on the time master of the second bus. Then the SL can just take the TUR value that is gained from the first bus and enforce this value for the second bus. In case the involved TTCAN controllers have the same interface, the whole rate synchronization essentially is a copying action.

5 Conclusion

We have demonstrated in this paper that there exist simple and in real-time applications usable mechanisms to synchronize all TTCAN busses of a fault-tolerant TTCAN network up to an extent that the whole fault-tolerant TTCAN network can be considered as one communication system. The range of supportable architectures goes far beyond redundant or partially redundant systems. The fault-tolerance properties of such a network are extremely strong as each TTCAN bus is a modular component that is not dependent on other channels of the system. Hence already by design the channels have some level of independence that allows “local” error management and reduces error propagation. A simple application of this principle allows to increase the effective data rate by using multiple TTCAN busses not for fault tolerance but just for higher data rate.

References

[1] Road Vehicles – Controller Area Network (CAN) – part 4: Time triggered communication, ISO CD 11898-4

[2] Timing in the TTCAN Network; F. Hartwich, T. Führer, R. Hugel, B. Müller, Robert Bosch GmbH; Proceedings 8th International CAN Conference; 2002; Las Vegas

Dr. Bernd Müller
Robert Bosch GmbH, FV/FLI
P.O. Box 106050
70049 Stuttgart, Germany
Tel. +49 711 811 7053
Fax. +49 711 811 7136
mueller.bernd@de.bosch.com

Thomas Führer
Robert Bosch GmbH, FV/FLI
P.O. Box 106050
70049 Stuttgart, Germany
Tel. +49 711 811 7597
Fax. +49 711 811 7136
thomas-peter.fuehrer@de.bosch.com

Florian Hartwich
Robert Bosch GmbH, AE/EIS
P.O. Box 1342
72703 Reutlingen, Germany
Tel. +49 7121 35 2594
Fax. +49 7121 35 1746
florian.hartwich@de.bosch.com

Robert Hugel
Robert Bosch GmbH, FV/SLN
P.O. Box 30 02 40
70049 Stuttgart, Germany
Tel. +49 711 811 8517
Fax. +49 711 811 1052
robert.hugel@de.bosch.com

Dr. Harald Weiler
Robert Bosch GmbH, FV/FLI
P.O. Box 106050
70049 Stuttgart, Germany
Tel. +49 711 811 7060
Fax. +49 711 811 7136
harald.weiler2@de.bosch.com