Timing in the TTCAN Network

Florian Hartwich, Bernd Müller, Thomas Führer, Robert Hugel, Robert Bosch GmbH

ISO TC22/SC3/WG1/TF6 has standardised (as ISO CD 11898-4) an additional layer to the CAN protocol, "Time Triggered Communication on CAN". This new standard specifies how a periodic transmission schedule is maintained, how a global system time is supported, and provides failure handling procedures as well as application interfaces.

The time triggered communication is built upon the unchanged CAN protocol (ISO 11898-1). This allows a software implementation of the time triggered function of TTCAN, based on existing CAN ICs. The high precision global time however requires a hardware implementation. A hardware implementation also offers additional functions like time mark interrupts, a stop-watch, and a synchronization to external events, all independent of software latency times.

This paper describes the principles of how a TTCAN network's clock speed and clock phase is synchronized to an external time base or to another TTCAN network.

INTRODUCTION

CAN is the dominating network for automotive applications. New concepts in automotive control systems (x-by-wire systems) require a time triggered communication. This is provided by TTCAN, ISO 11898-4 [2]. The main features of TTCAN are the synchronization of the communication schedules of all CAN nodes in a network, the possibility to synchronize the communication schedule to an external time base, and the global system time.

TTCAN nodes are fully compatible with CAN nodes, both in the data link layer (ISO 11898-1 [1]) and in the physical layer; they use the same bus line and bus transceivers. Dedicated bus guardians are not needed in TTCAN nodes, bus conflicts between nodes are prevented by CAN's non-destructive bitwise arbitration mechanism and by CAN's fault confinement (error-passive, bus-off).

Existing CAN controllers can receive every message in a TTCAN network, TTCAN controllers can operate in existing CAN networks. A gradual migration from CAN to TTCAN is possible.

The minimum additional hardware that is required to enhance an existing CAN controller to time triggered operation is a local time base and a mechanism to capture the time base, the capturing triggered by bus traffic. Based on this hardware, which is already existent in some CAN controllers, it is possible to implement in software a TTCAN controller capable of TTCAN level 1. A TTCAN controller capable of TTCAN level 2, providing the full range of TTCAN features like global time, time mark interrupts, and time base synchronization, has to be implemented in silicon.

A TTCAN controller can be seen as an existing CAN controller (e.g. Bosch's C_CAN module) enhanced with a Frame Synchronization Entity FSE and with a trigger memory containing the node's view of the system matrix (see Figure 1).

The TTCAN testchip (TTCAN_TC) is a standalone TTCAN controller and has been produced as a solution to the hen/egg problem of hardware availability versus tool support and research. The TTCAN_TC supports both TTCAN level 1 and TTCAN level 2; its time triggered communication is not depending on software control. All synchronization mechanisms described in this paper are supported by the TTCAN_TC.

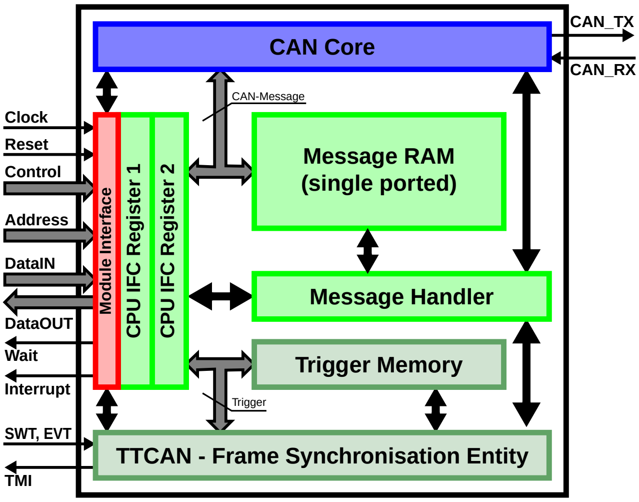

Figure 1: TTCAN Controller Module

1 Time Bases of the TTCAN Protocol

Local_Time

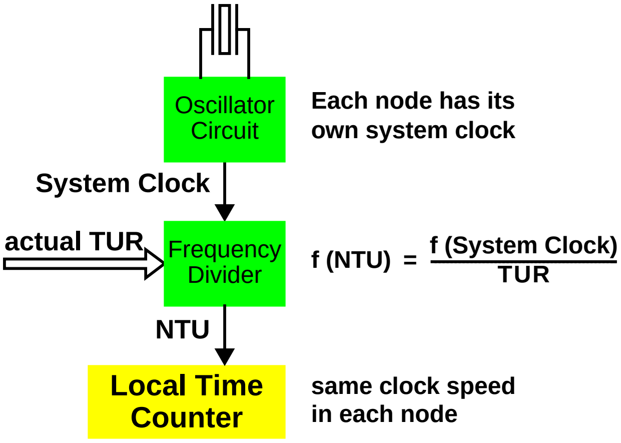
Oscillator Circuit
Each node has its own system clock
System Clock
actual TUR
Frequency Divider
f (NTU) = f (System Clock) / TUR
NTU
Local Time Counter
same clock speed in each node

Figure 2: Local Time

Each node has its own time base, Local_Time, which is a counter that is incremented each Network Time Unit NTU. The length of the NTU is defined by the TTCAN network configuration, it is the same for all nodes. It is generated locally, based on the local system clock period t_{sys} and the local Time Unit Ratio TUR, $\mathrm{NTU} = TUR \bullet t_{sys}$. Different system clocks in the nodes require different (non-integer) TUR values.

In TTCAN level 1, TUR is a constant and Local_Time is a 16 bit integer value, incremented once each NTU. The NTU is the CAN bit time.

In TTCAN level 2, Local_Time consists of a 16 bit integer value extended by a fractional part of N (at least three) bit. Local_Time is incremented 2^N times each NTU, providing a higher time resolution than in level 1. TUR is a non-integer value and may be adapted to compensate clock drift or to synchronize to an external time base.

Cycle_Time

In the TTCAN network, the synchronization of the nodes is maintained by so-called Reference Messages that are transmitted periodically by a specific node, the time master. The Reference Message is a CAN data frame, characterized by its identifier. Valid Reference Messages are recognized synchronously (disregarding signal propagation time) by all nodes. Each valid Reference Message starts a new basic cycle and causes a reset of each node's Cycle_Time.

The value of Local_Time is captured as Sync_Mark at the start of frame (SOF) bit of each message. When a message is recognized as a valid Reference Message, this message's Sync_Mark becomes the new Ref_Mark; Cycle_Time is the actual difference between Local_Time and Ref_Mark, restarting at the beginning of each basic cycle when Ref_Mark is reloaded.

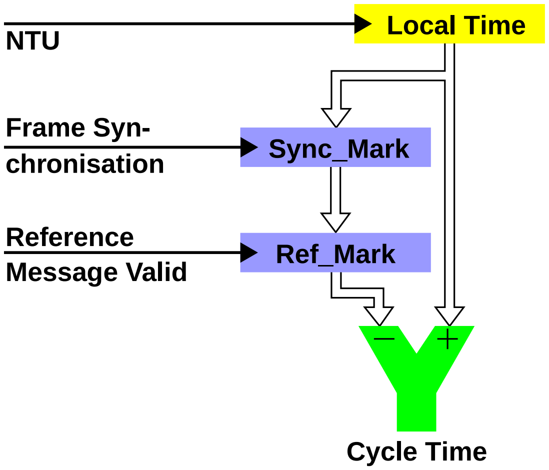

Figure 3: Cycle Time

Even in a software implementation of TTCAN, the capturing of Local_Time into Sync_Mark at each SOF must be done in hardware (see Figure 2). ISO 11898-1 [1] specifies the necessary hardware interface as an optional feature, it is already implemented in some CAN controllers.

Global_Time

There are two levels of implementation in TTCAN, level 1 and level 2. In TTCAN level 1, the common time base is the Cycle_Time which is restarted at the beginning of each basic cycle and is based on each node's Local_Time. In TTCAN level 2, there is additionally the Global_Time which is a continuous value for the whole network and is the reference for the calibration of all local time bases.

The time master captures its view of Global_Time at each Sync_Mark and transmits that value in the Reference Message, as Master_Ref_Mark. For all nodes, Global_Time is the sum of their Local_Time and their Local_Offset, Local_Offset being the difference between their Ref_Mark in Local_Time and the Master_Ref_Mark in Global_Time, received (or transmitted) as part of the Reference Message. The Local_Offset of the current time master is zero if no other node has been the current time master since network initialization.

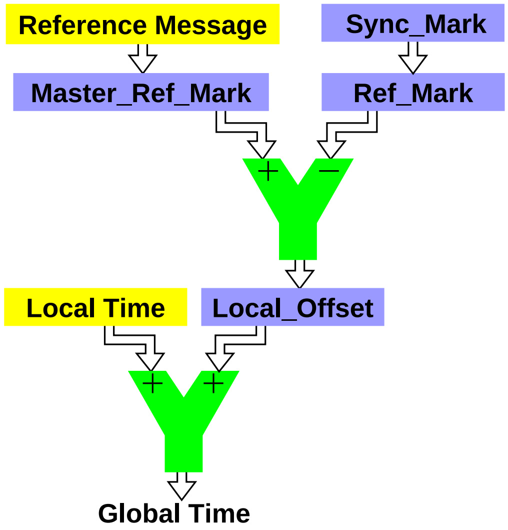

Figure 4: Global Time

The phase drift between Local_Time and Global_Time is compensated at each received Reference Message by updating Local_Offset.

Changes in Local_Offset show differences in the local node's NTU and the actual time master's NTU. The actual clock speed difference is calculated by dividing the differences between two consecutive Master_Ref_Marks (measured in global NTUs) and two consecutive Ref_Marks (measured in local NTUs). The clock speed drift is compensated by adapting the prescaler (TUR) that generates the local NTU from the local system clock (see Figure 5).

The factor df by which the local NTU has to be adjusted is calculated according to the formula:

$$df = \frac{\text{Ref\_Mark} - \text{Ref\_Mark}_{previous}}{\text{Master\_Ref\_Mark} - \text{Master\_Ref\_Mark}_{previous}}$$

$$TUR = df \bullet TUR_{previous}$$

The calibration process is on hold when the node is not synchronized to the system, it is (re-)started when it (re-)gains synchronization. The necessary accuracy of the calibration is defined by the system's requirement, a plausibility check for the value of df ensures that the length of the NTU remains in a predefined range.

This calibration, together with the higher resolution for the NTU, provides a high precision time base.

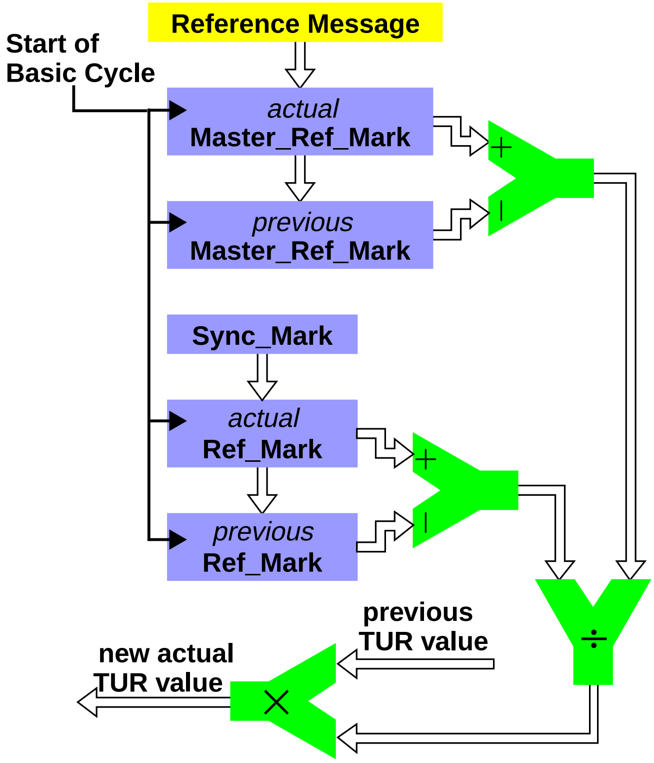

Figure 5: Drift Compensation

After initialisation, before synchronizing to the network, each node sees its own Local_Time as Global_Time, the Local_Offset is zero. The actual time master establishes its own Global_Time as the network's Global_Time by transmitting its own Global_Sync_Marks in the Reference Message, as Master_Ref_Marks. When a backup time master becomes the actual time master, it keeps its Local_Offset value constant, avoiding a discontinuity of Global_Time.

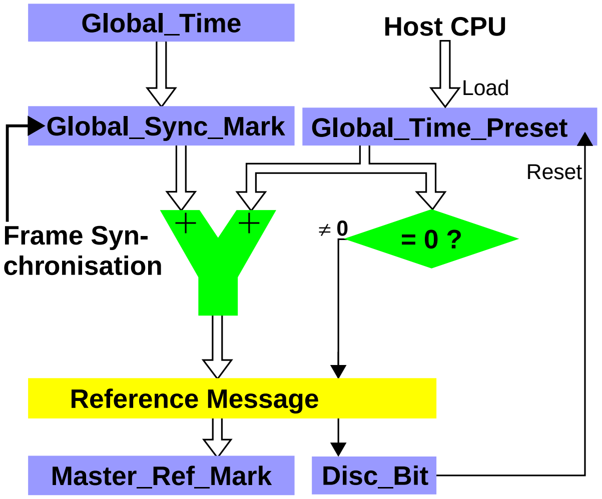

Figure 6: Global Time Phase Adjustment

2 Synchronizing the Global_Time

When the TTCAN communication is initialized, the actual time master may adjust the phase of Global_Time by adding an offset (Global_Time_Preset, see Figure 6) to the transmitted Master_Ref_Mark value, e.g. to synchronize to an external clock. Any such intended discontinuity of Global_Time is signalled in the Reference Message, by setting the Disc_Bit. Reference Messages with a set Disc_Bit are not used for clock calibration.

The actual time master may adjust the speed of Global_Time by adjusting its TUR value, the other nodes in the TTCAN network will calibrate their own clocks as shown in Figure 5.

The external time base used for the synchronization of Global_Time may be a reference clock like GPS or the Global_Time monitored in another TTCAN network.

3 Synchronizing the Cycle_Time

TTCAN has the option to synchronize the communication schedule to specific events in the time masters' nodes. When the communication is to be synchronized, the cyclic message transfer is discontinued after the end of a basic cycle and a time gap may appear between the end of that basic cycle and the beginning of the next, event synchronized basic cycle. The current time master announces the time gap by setting the Next_is_Gap bit in the Reference Message. The time gap ends as soon as the current time master or one of the potential time masters sends a Reference Message to start the following basic cycle of the matrix cycle. The transmission of the Reference Message will be triggered by the occurrence of a specific event or after a maximum waiting time.

Time Schedule Organizer – TSO

This block is a state machine that maintains the message schedule inside a basic cycle. The TSO gets its view of the message schedule from an array of time triggers in the trigger memory. Each time trigger has a time mark that defines at which Cycle_Time the trigger becomes active.

A Tx_Trigger specifies when a specific message shall be transmitted.

An Rx_Trigger specifies when the reception of a specific message shall be checked.

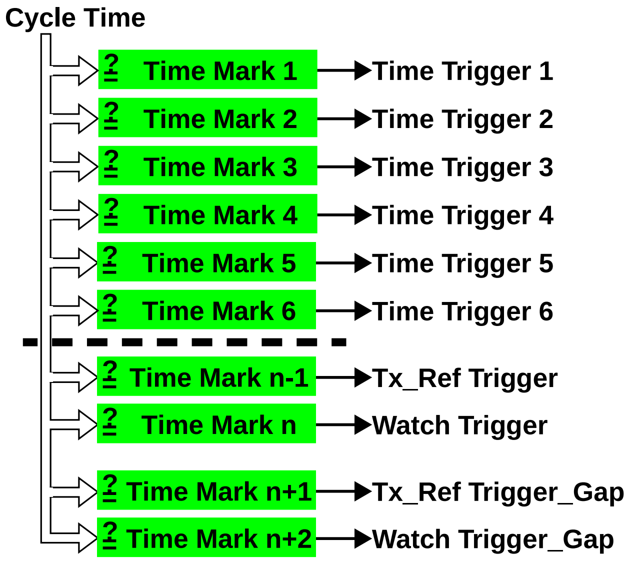

Figure 7: Time Trigger

A Tx_Ref_Trigger(_Gap) triggers the transmission of a Reference Message, it finishes the current basic cycle and starts a new cycle. Ref_Triggers are used by potential time masters only.

A Watch_Trigger(_Gap) has a Time_Mark with a higher value than the Tx_Ref_Trigger(_Gap) and checks if the time since the last valid Reference Message has been too long.

When in the last Reference Message the Next_is_Gap bit was set, the TSO ignores Tx_Ref_Trigger and Watch_Trigger, it uses Tx_Ref_Trigger_Gap and Watch_Trigger_Gap instead. In all other cases, Tx_Ref_Trigger and Watch_Trigger are used, Tx_Ref_Trigger_Gap and Watch_Trigger_Gap are ignored.

The maximum time allowed for a time gap is the difference Tx_Ref_Trigger_Gap - Tx_Ref_Trigger.

Host controlled Synchronisation

Figure 8 shows an example how the host application of the time master can synchronize the TTCAN network's Cycle_Time. First the host request the time master to transmit a Reference Message with the Next_is_Gap bit set. The time gap starts when the basic cycle started by that reference Message is finished. The message schedule is restarted when the host triggers the next Reference Message. If the host fails to trigger the Reference Message within a specified time, the TSO itself triggers the Reference Message when its Cycle_Time reaches Tx_Ref_Trigger_Gap.

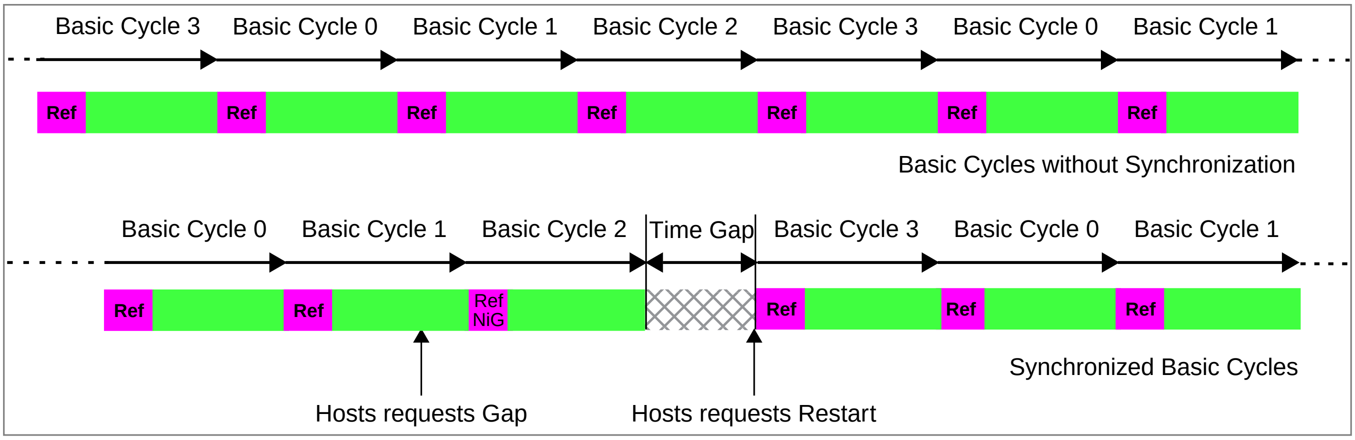

Figure 8: Host Synchronization of Cycle Time Phase

Automatic Synchronization

The implementation of TTCAN in hardware allows to implement some additional features (not required by TTCAN protocol) that cannot be provided in software.

An Event Trigger input (EVT, see Figure 1) can be used to trigger Reference Messages. In this mode, the time master transmits each Reference Message with Next_is_Gap bit set. The input level at the time master's EVT pin controls the time gap: When EVT is high at the end of a basic cycle, a time gap is started. The Reference Message to end the time gap is triggered at the next falling edge of EVT (see Figure 9). If the falling edge does not occur within a specified time, the TSO itself triggers the Reference Message when its Cycle_Time reaches Tx_Ref_Trigger_Gap.

No time gap is started when EVT is low at the end of a basic cycle; only falling edges that occur during a time gap can trigger a Reference Message.

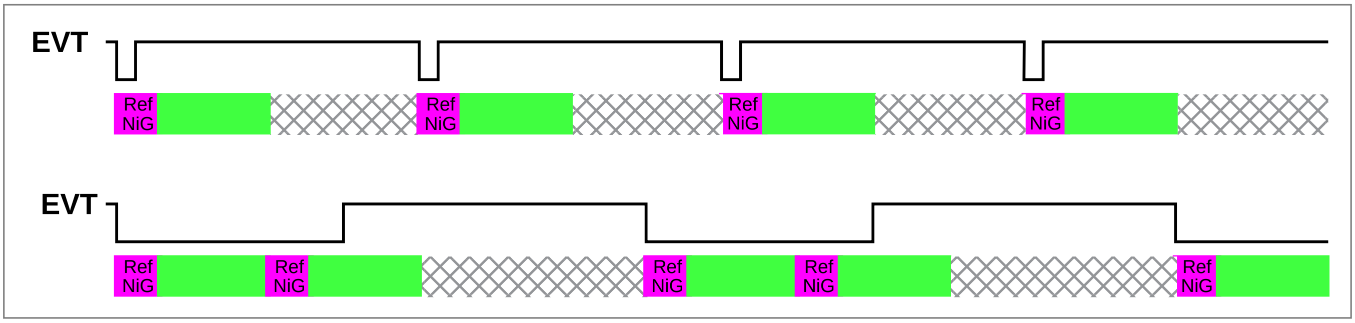

Figure 9: Automatic Synchronization of Cycle Time Phase

4 Time Measurement in TTCAN

In TTCAN level 1, there are two time bases, the Local_Time and the Cycle_Time. In level 2, there is additionally Global_Time. The host application has read access to all time bases, it can store the actual time value read at specific events, e.g. controlled by an interrupt service routine.

A hardware implementation of TTCAN permits some features that are not possible in a software implementation, like bus-time-based interrupts, a stop-watch function, and the event trigger EVT.

Time Mark Interrupt

Local_Time, Cycle_Time, and Global_Time can be compared to a time mark interrupt register. When the selected time value matches the register value, an interrupt is generated. This event may trigger the CPU's interrupt line or may be directly connected to an output port (TMI, see Figure 1). The TMI output(s) can be used to synchronize the application to the TTCAN's Cycle_Time or Global_Time.

Stop-Watch

An input port (SWT, see Figure 1) may be used to trigger the capturing of Local_Time, Cycle_Time, or Global_Time to a stop-watch register. Once it has been triggered, the stop-watch register remains unchanged until the host CPU has read its contents and has enabled the next triggering. This allows the clocking of events in a TTCAN network time base without interrupt response time jitter and without CPU load.

5 Synchronization of several TTCAN Networks

Time mark interrupt, stop-watch, and event trigger can be used for the synchronization between application and TTCAN network as well as for the synchronization between different TTCAN networks.

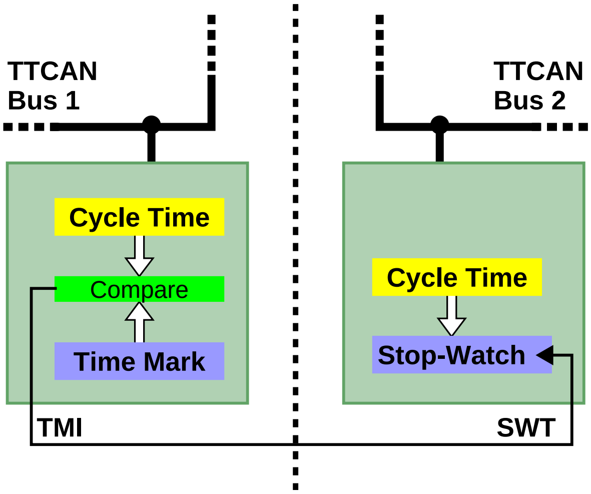

Figure10:Basic Cycle Phase Measurement

When a node is connected to more than one TTCAN network, e.g. as a gateway node with two TTCAN controllers, it can measure the differences in the clock speed and in the phases of Cycle_ Time and Global_Time by connecting the time mark interrupt output (TMI) of one TTCAN controller to the stop-watch input (SWT) of the other TTCAN controller (see Figure 10). The difference between the time mark interrupt register in the TTCAN controller connected to TTCAN network 1 and the stop-watch register in the TTCAN controller connected to TTCAN network 2 shows the phase shift between the two communication schedules.

When this TTCAN node is time master in one of the networks, it can synchronize the two communication schedules by triggering the time master's EVT input with the TMI output of the other TTCAN controller (see Figure 11).

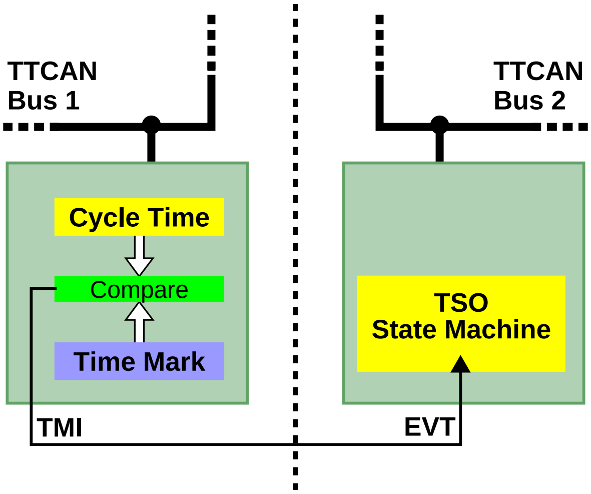

Figure11:Communication Schedule Synchronization

It is not necessary that both TTCAN networks operate with the same basic cycle length; they may use different cycle lengths and may operate on different CAN bit times.

A time master can adjust the network's clock speed by modifying its actual TUR value. Figure 12 shows an example where the node is time master in network 2 and calibrates its clock to the same speed as the NTU in network 1. In TTCAN level 2, the other nodes in the same network will automatically synchronize their local clock speeds to the time master's clock speed. In TTCAN level 1, there is no automatic clock speed synchronization.

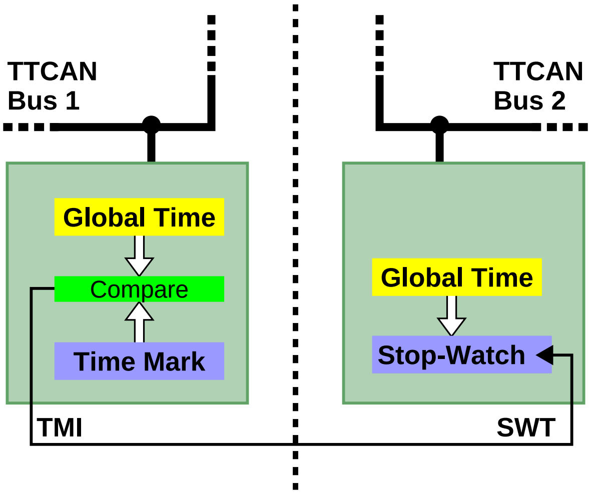

Figure12:Clock Synchronization of TTCAN Networks

When a node is time master in one of the TTCAN networks, it can adjust clock speed and clock phase of that network until the synchronization with the other network is achieved. When the gateway node is not time master, it transmits the results of the measurements to the time master that will perform the synchronization.

Any number of TTCAN networks that are connected by (a chain of) gateway nodes can be synchronized that way, providing a common Global_ Time for the combined fault tolerant system [3].

6 Conclusion

The TTCAN protocol provides several synchronization features, not only the calibration of the clock speed inside the TTCAN network, but also synchronization between application and TTCAN network and synchronization between different TTCAN networks.

The implementation of the full range of TTCAN features, including the global time and clock calibration, requires a dedicated TTCAN controller in hardware. An implementation of TTCAN in software (level 1 only) based on an ISO 11898-1 CAN controller cannot provide the additional synchronization interfaces like TMI, SWT, and EVT.

This paper shows some examples of how the synchronization interfaces of a TTCAN controller implemented in hardware can be used to synchonize communication cycles, Global_Time, and clock speeds between two TTCAN networks.

In parallel to the ISO standardization process, Bosch has implemented the TTCAN protocol (level 1 and level 2) into a CAN IP module [4] that performs all protocol functions in logic, not depending on software control and that provides time mark interrupts, a stop-watch, and an event trigger input. The module, synthesized into FPGAs, was used for research applications in support of the standardization.

The TTCAN IP module has also been implemented in silicon for evaluation purposes. The IC is available as samples in PLCC44 package and is called TTCAN_TC, the TTCAN testchip.

The TTCAN_TC supports both TTCAN level 1 and TTCAN level 2, its package is pin-compatible to existing standalone CAN controllers (Intel 82527 / Bosch CC770). It is currently used in the design of the first TTCAN network planning and analysis tools.

References

1. ISO/DIS 11898-1; Road vehicles – Controller area network (CAN) – Part 1: Controller area network data link layer and medium access control.
2. ISO/CD 11898-4; Road vehicles – Controller area network (CAN) – Part 4: Time triggered communication.
3. Fault Tolerant TTCAN Networks; B. Müller, T. Führer, F. Hartwich, R. Hugel, H. Weiler, Robert Bosch GmbH; Proceedings 8th International CAN Conference; 2002; Las Vegas.
4. TTCAN User’s Manual; Robert Bosch GmbH; 2001; http://www.can.bosch.com/docu/Users_Manual_ TTCAN.pdf.

Contact

Florian Hartwich
Robert Bosch GmbH, AE/EIS
P.O.Box 13 42
72703 Reutlingen
Germany
Phone: +49 7121 35-2594
Fax: +49 7121 35-1746
E-mail: Florian.Hartwich@de.bosch.com

Thomas Führer
Robert Bosch GmbH, FV/FLI
P.O.Box 10 60 50
70049 Stuttgart
Germany
Phone: +49 711 811-7597
Fax: +49 711 811-7136
E-mail: Thomas-Peter.Fuehrer@de.bosch.com

Dr. Bernd Müller
Robert Bosch GmbH, FV/FLI
P.O.Box 10 60 50
70049 Stuttgart
Germany
Phone: +49 711 811-7053
Fax: +49 711 811-7136
Mail: Mueller.Bernd@de.bosch.com

Robert Hugel
Robert Bosch GmbH, FV/SLN
P.O. Box 30 02 40
70442 Stuttgart
Germany
Tel: +49 711 811 8517
Fax: +49 711 811 1052
E-mail: Robert.Hugel@de.bosch.com

Additional Sources

http://www.can.bosch.com/content/TT_CAN.html

Definitions, Acronyms, Abbreviations

CAN:	Controller Area Network
EVT:	Event Trigger Input
FSE:	Frame Synchronization Entity
FPGA:	Field Programmable Gate Array
IC:	Integrated Circuit
IP:	Intellectual Property
NiG:	Next_is_Gap
NTU:	Network Time Unit
PLCC:	Plastic Leaded Chip Carrier
SOF:	Start of Frame
SWT:	Stop-Watch Trigger Input
TMI:	Time Mark Interrupt Output
TSO:	Time Schedule Organizer
TTCAN:	Time Triggered CAN
TUR:	Time Unit Ratio