MultiCAN – a step to CAN and TTCAN

Ursula Kelling, Infineon Technologies

MultiCAN is a new CAN module, which is building up a bridge between TTCAN level 1 & 2 and normal CAN network. On one hand it has advanced features to mould the system matrix within the build-in scheduler on the other hand it has an improved list feature, a hardware gateway and additional features for CAN diagnostics. This paper compares and shows the improvement between the TwinCAN implementation and the new MultiCAN implementation. Finally it shows how to build up the system matrix and how to use the alternative message, in case the information for exclusive time windows is not ready.

MultiCAN – An Introduction

MultiCAN is a new scalable module, which will become the new Infineon family CAN. This module supports from one, up to eight CAN nodes, with one of them fully hardware supporting TTCAN (level 1 & 2). The number of message objects can be up to 256 message objects, shared for all nodes. The current implementation supports four independent nodes with 128 message objects fully CAN 2.0b compliant and one node TTCAN level 1 and 2 compliant. All nodes support an analyzer feature, which allows to connect to the bus system as a passive member of the bus. Message objects can be concatenated form of a FIFO or to a hardware supported gateway.

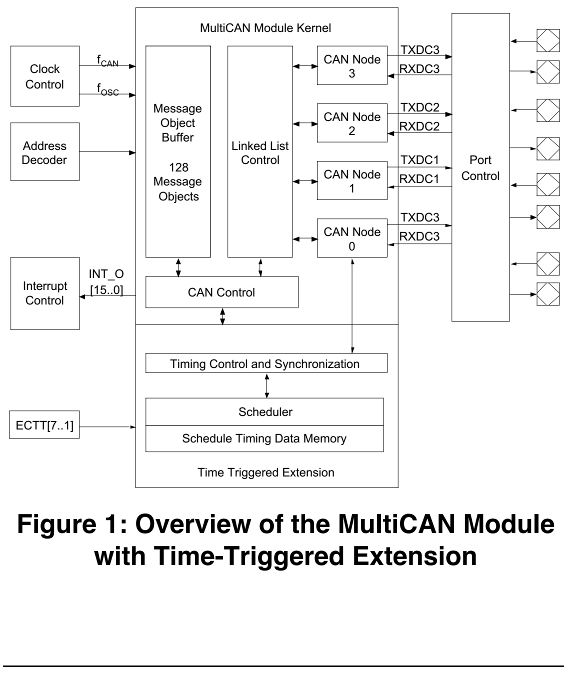

Figure 1: Overview of the MultiCAN Module with Time-Triggered Extension

TTCAN and CAN nodes

All nodes support CAN, but one node additionally supports TTCAN level 1 and 2 by hardware. This node offers a built in scheduler to support the system matrix. For each message object 'connected' to the TTCAN system, an emergency message (alternative message) is supported. The alternative message can be sent, if in an exclusive time window the information for this time window is not ready. Additionally to these features, global timing is available and a timing related interrupt, which can wake up the system, e.g. out of the gap mode.

Differences: TwinCAN – MultiCAN

MultiCAN as a successor of TwinCAN, gives advanced features. On TwinCAN only two nodes were possible. Now more nodes are supported as before a shared message object memory part are possible to give a flexible environment. To provide the flexibility in terms of number of message objects per node, a single bit had to be set, to provide the node information. Now, having more nodes and additionally double chained lists a bitfield became necessary. The list (node) information can be found in every message object, to give a decision parameter for the CAN controller and an overview of the actual position of every message object for the user. In Figure 2 an example for this list – node connection is given.

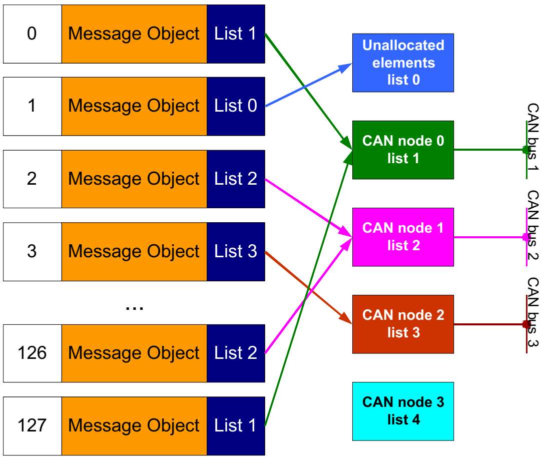

Figure 2: Mapping Message objects on CAN nodes

As already on TwinCAN each message object has a local mask to give the possibility of receiving a group of message identifiers. Every position of a '0' inside the mask is a don't care for the position in the identifier received from the bus. The flexibility of the message objects in terms of FIFO has been increased, by replacing the fixed FIFO structure, by a list structure, featuring a double-chained list over several 'collected' message objects. An example for a list over message objects 5, 16 and 3, is given in Figure 3.

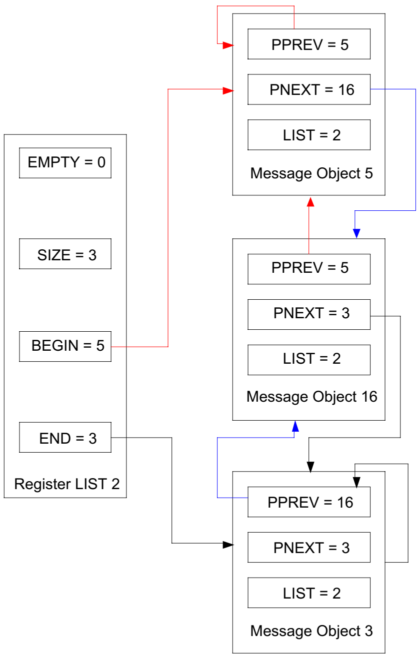

Figure 3: List over message object 5 - 16 - 3

As with TwinCAN a hardware gateway feature has been implemented. This hardware gateway gives the possibility to forward messages from one bus system to another that possibly run may on different speeds. Using gateways is quite interesting to connect different subsystems in a vehicle; an application is shown in Figure 4. This feature runs without any additional CPU load at all.

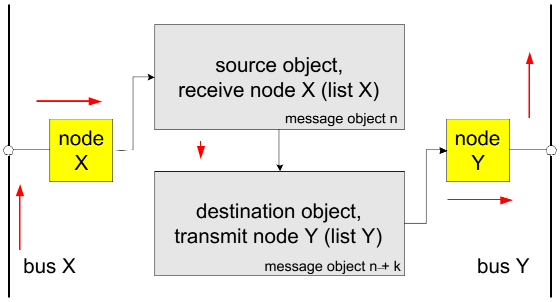

Figure 4: Principal of a gateway

The combination of gateway and list feature is possible. For example on a high speed bus system a high frequent message shall be forwarded to a low speed bus, here the combination of list and gateway features arises the possibility of forwarding those messages without any additional CPU load.

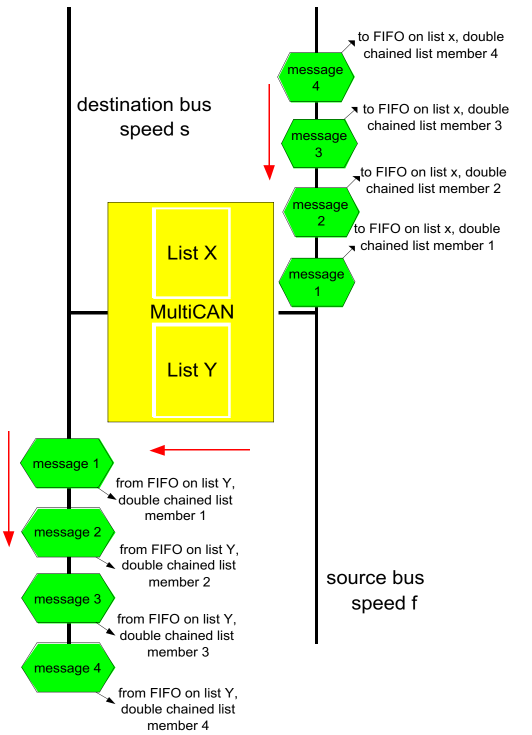

Figure 5: Sample for a combined list feature - gateway application without CPU support

TTCAN

To make the step to TTCAN, on MultiCAN a TTCAN unit has been added, to support TTCAN level one and two by hardware.

Global and local time of a TTCAN network is calculated automatically, including the information for the TUR (Time Unit Ratio) adjust value. The TUR adjust value is calculated after the correct reception of a reference message. TTCAN is fully hardware assisted on node '1'. Beside a hardware scheduler, to support the system matrix of a TTCAN system providing the exact position and is fully write protected during runtime, the module can be used as a time master of level 1 or 2 TTCAN system. Local and cycle time are available for the application, the global time can be calculated. In case the user needs the next_is_gap mode (wake up via event), the wakeup event can be defined on timing information or on a trigger signal on predefined pins.

Unique Benefits of MultiCAN

MultiCAN offers a bouquet of unique benefits. First of all, the hardware scheduler gives the possibility of mapping the system matrix onto the scheduler by inserting the involved time marks. Focusing on exclusive time windows, the TTCAN protocol itself gives features to detect, that a message has not been sent. MultiCAN offers additionally a feature to send a predefined message in case the message for exclusive time window is not ready, due to the fact, that the information hasn't been ready on time. In a system, where TTCAN level 2 allows the next_is_Gap feature, a timing interrupt can take over the wake up event. In any case, the global time is readable for applications.
As already on TwinCAN the so-called analyzer mode exits. In analyzer mode the CAN node can listen to the bus, without participating. This feature is valuable for example, in case the baudrate shall be detected, without disturbing the bus.
New to the module is a feature, which enhances the user or an application to detect, whether the resistor on the bus is the right termination. This feature gives the possibility to calculate the distance, between an outgoing edge and its runtime back over the receive pin.

MultiCAN features: CAN Debugging Features

To avoid errors and to ensure to use the right error-handling MultiCAN offers a variety of mechanisms.
In case the device has two CAN modules, the following possibilities exist:

Two CAN modules, two CAN transceivers

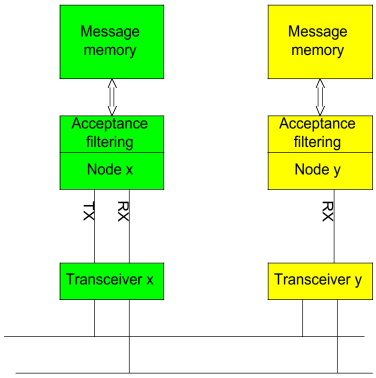

Figure 6: Two separate CAN nodes, two transceivers

Minus: This concept needs two CAN transceivers
Plus: Transceiver errors can be detected

Two CAN modules, one CAN transceiver

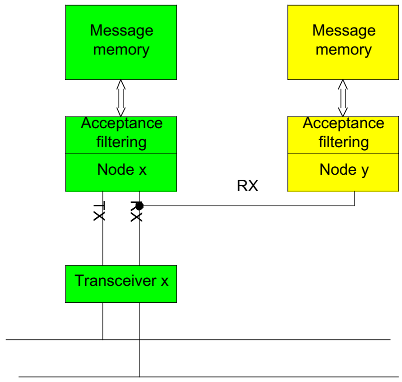

Figure 7: Two separate CAN nodes, one transceiver

Plus: This concept needs one CAN transceiver.
Minus: A transceiver error cannot be detected.

Both concepts do have synchronized message access, which basically leads to the fact that short time disturbances are recognized by both CAN nodes at the same time and lead to an error situation.

MultiCAN in Analyzer Mode
In Analyzer Mode the device listens to the bus, but it is not actively taking part of the protocol. The nodes are handled

asynchronously as the protocol handler is requesting the information sequentially. Message objects are assigned to each node. If the application shall test, that a received message is really correct, for each node a message object has to be configured and compared via software. Disturbances on the bus are seen in different states and have a different impact. It is even possible to find a problem between protocol handler and message memory.

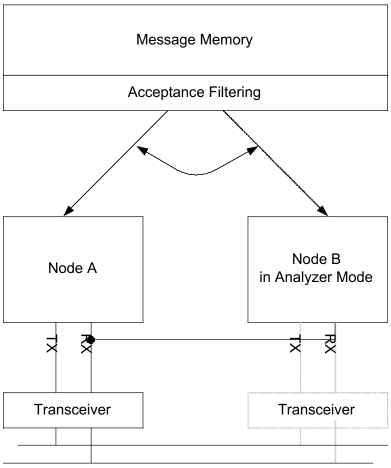

Figure 8: MultiCAN in Analyzer Mode

Beside the Analyzer Mode the MultiCAN module has additional features to avoid bus errors before they occur. The MultiCAN offers a synchronization analysis features as well as driver delay measurement feature, which help to make further extensions of the bus safer and small disturbances on the bus less critical.

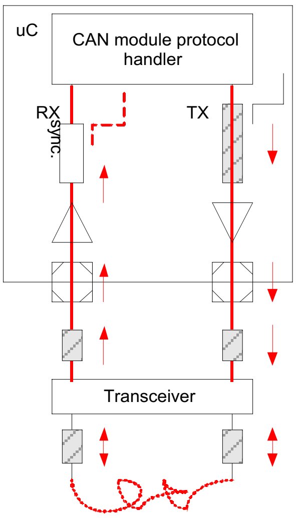

Figure 9: Schematic signal way, TX to RX

The synchronization analysis feature helps to prevent a possible bus error in case of slight changes or impacts on the bus. This feature monitors the time between the first dominant edge and the sample point measured a stored in a frame counter register. By using this feature it is easier to adjust the sample point to the actual sample point of the complete bus system.
The driver delay measurement feature helps to find a circuitry, where a sent edge is received time quanta before the sample point.
Both features help to find a configuration and circuitry, which is not fault induced by to slide changes or disturbances on the bus.

MultiCAN features: Inbuilt Scheduler

The system matrix is handled by a scheduler mechanism. Each time mark can be placed within the scheduler memory. This scheduler is based on the cycle time and delivers the programmed time marks to the TTCAN system. The time marks are defined by the entry points of each time window. Whenever a time mark is reached, programmable actions can take place, triggered by the inbuilt scheduler.
Typical instructions are for example:

- starting the transmission of a message,
- check if a message has been received,
- open or close arbitration windows,
- or generate interrupts.

The instructions following a time mark are read by the scheduler until the next time mark entry is found. Afterwards the instruction collection process is suspended until the next time mark is reached. As default status the scheduler is set on exclusive time window. To change the status back to exclusive time window, after a sequence of merged arbitrating windows, a single (short) arbitrating window has to be programmed.

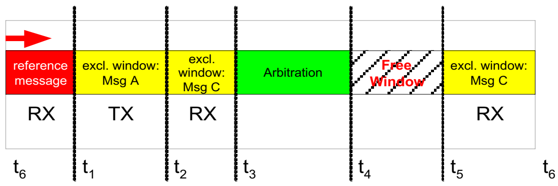

Figure 10 : Single basic cycle of a time slave

The scheduler knows eight different types of events (entries), which can be defined:

- Time mark,

- interrupt control,
- arbitration,
- transmit control,
- receive control,
- reference message,
- basic cycle end
- and end of scheduler memory.

On each time mark, it is possible to generate an interrupt. A special interrupt is the interrupt on the reference message. This interrupt can trigger actions on other modules or give a regular timer event for the system. In the example of Figure 10, the TTCAN node shall transmit Message A and receive message C.

command	Arbitration mode	Interrupts	Event time
timer mark	No action	Here: none	t1
command	Check enable	Destination RX MO of interest	1st basic cycle / repeat factor
RX control	Yes	MO of ref msg	0 / 1
command	Alternative msg.	Transmission enabled	1st basic cycle / repeat factor
TX control	No msg., if destination object not valid.	Next TX trigger sends MO of msg A.	0 / 1
command	Arbitration mode	Interrupts	Event time
timer mark	No action	Here: none	t2
command	Arbitration mode	Interrupts	Event time
timer mark	No action	Here: none	t3
command	Check enable	Destination RX MO of interest	1st basic cycle / repeat factor
RX control	Yes	MO of msg. C	0 / 1
command	Arbitration mode	Interrupts	Event time
timer mark	short arbitration window	Here: none	t4
command	Arbitration mode	1st basic cycle / repeat factor	
Arbitration	short arbitration window	0 / 2	
command	Arbitration mode	Interrupts	Event time
timer mark	no action: excl time window	Here: none	t5
command	Gap mode	Event time	
basic cycle end	No	t6	
command	Arbitration mode	Interrupts	Event time
timer mark	No action	Here: none	t6

command	Arbitration mode	Interrupts	Event time
command	Check enable	Destination RX MO of interest	1st basic cycle / repeat factor
RX control	Yes	MO of msg. C	0 / 1

Figure 11: Scheduler example for Figure 10

The scheduler example in Figure 11 shows a way, how a schedule is defined on MultiCAN. Additionally, the scheduler has an error detection mechanism, which helps to find, the following configuration errors:

- At the end of basic cycle, an arbitration window is still open.
- A slave device has a reference message scheduled.
- A time mark is missing during instruction collection.
- In a reference message window, another message window type is scheduled.
- A reference message is missing.

As all events refer to a time mark this schedule has a deterministic behavior. Only inside arbitration windows, it is not known, which message will be sent on the bus before the complete system is defined.

MultiCAN features: Alternative Message

A further feature that is not used in this schedule example is the so-called alternative message. In exclusive time windows, it is possible to transmit a scheduled message or – if that is not available - a message that is triggered by another object. The user has the possibility to configure, the message, which shall have higher priority. For example the message inside scheduler memory has higher priority. The exclusive time window starts and this message is not ready. In this case the user is sending the message of the message object, which has been defined as backup. Maybe the user likes to get noticed in case measurements are not ready; here an alternative message can be sent.

MultiCAN features: Flexible Interrupts

MultiCAN supports up to 16 interrupt nodes routable on different events (service request nodes). Each CAN node has four different interrupt events. Each message object can trigger an interrupt. The frame counter, available for frame or timing information has an overflow service request. The TTCAN scheduler has different service requests to trace the complete TTCAN traffic for this node. The principle of the interrupt node pointer selection mechanism is shown in Figure 12. Each event can be routed on one of the interrupt nodes. The interrupt number for every service request event is routable by writing the interrupt number to a bitfield.

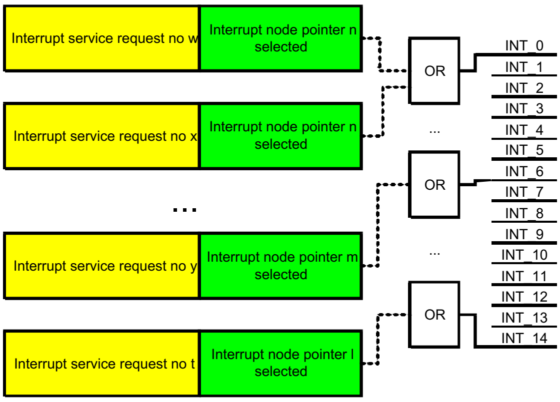

Figure 12: Principle of the interrupt node pointer selection

Integration into a TTCAN network

After a reset, the TTCAN extension hast to be configured. The configuration mode is entered automatically after reset or can be initiated through software by writing a one to a register, called TTFMR. The configuration mode can only be left, by ORing a two to this register. A status flag indicates whether the configuration mode is still active or not. During the configuration mode of the TTCAN node, the following actions have to be done by software or are executed by hardware:

- The local time, the global time, and the cycle time are set to 0 (hardware).
- The transmission or reception of messages of TTCAN node is not possible, because the results of the acceptance filtering are not enabled (hardware).
- An appropriate Time Unit Ratio value must be written to bit field TUR adjust (software). This value is automatically transferred to TUR (hardware).
- The scheduler memory entries must be initialized (software).
- The TTCAN control information (ID of reference message, etc.) and the TTCAN node itself must be set up completely and enabled for CAN message transfer (software).
- After the complete configuration (by software), TTFMR must be ORed with two (software).
- The local time starts after leaving the configuration mode (hardware) and the synchronization phase is entered automatically indicated by the SYNC state (hardware).

In case of a potential master node, the node starts sending the reference message, otherwise the node wait until leaving the SYNC state (two times reference message). After leaving the synchronization state, the scheduler is used as programmed.

Conclusion

MultiCAN is a FullCAN module integrating all advantages of high flexibility. The module is TTCAN compliant, by not lacking the CAN compliance. In case TTCAN is used it gives a complete hardware support for the system matrix. In short, MultiCAN is a bridge between CAN and TTCAN.

Author: Ursula Kelling
Company: Infineon Technologies
Address: Balanstr. 73
81541 Munich
Germany
Phone: +49 (89) 234 83287
Fax: +49 (89) 234 955 6811
E-mail: Ursula.Kelling@infineon.com
Website: www.infineon.com