CAN with Flexible Data-Rate

Florian Hartwich, Robert Bosch GmbH

Ever increasing bandwidth requirements in automotive networks impede the applicability of CAN due to its bit rate limitation to 1 MBit/s. To close the gap between CAN and other protocols, we improve CAN in two ways: (i) support of bit rates > 1 MBit/s and (ii) support of payloads > 8 byte per frame. We achieve this with a new frame format where we can switch inside the frame to a faster bit rate for (i) and use a different data length coding for (ii). This new protocol is called "CAN with Flexible Data-Rate" or CAN FD. CAN FD protocol controllers are also able to perform standard CAN communication. This allows to use CAN FD in specific operation modes, e.g. software-download at end-of-line programming, while other controllers that do not support CAN FD are kept in standby.

This paper presents the CAN FD frame format with additional bits in the control field to enable the new options and the new CRC sequence to secure longer frames with the same Hamming distance as in the existing CAN protocol. The configuration options for the two bit rates are explained in detail. We provide measurements of the upper limits for the bit rate, using the first hardware implementation of a CAN FD protocol controller and standard CAN transceivers.

1. Introduction

Increasing system complexity can fill a CAN network's communication bandwidth to its limit. Solving this problem by using multiple CAN buses or by switching to another protocol requires high effort in system design as well as replacing hardware and software.

Over the years, several concepts have been proposed how to replace CAN with new bus systems that have a higher bandwidth and a similar controller host interface. The similarity avoids the need for major software modifications.

CAN's bandwidth limit is closely linked to one of its greatest advantages, its non-destructive arbitration mechanism for media access control. This mechanism requires that the signal propagation time between any two nodes is less than half of one bit time and so defines an upper boundary for the bit rate as well as for the bus length.

Therefore, new concepts to increase the CAN bit rate avoid this limit mainly by two alternatives. Firstly, the authors in [3] [5] [10] change CAN's multi-master bus line to a star (or tree) topology where arbitration (or even message routing) is performed inside an active star. Secondly, the authors in [2] [4] [6] use two alternate bit rates and switch - after the arbitration - from the lower to the higher bit rate.

CAN FD [1] has been developed with the goal to increase the bandwidth of a CAN network while keeping most of the software and hardware - especially the physical layer – unchanged. Consequently, only the CAN protocol controllers need to be enhanced with the CAN FD option. The new frame format makes use of CAN's reserved bits. Via these bits, a node can distinguish between the frame formats during reception. CAN FD protocol controllers can take part in standard CAN communication. This allows a gradual introduction of CAN FD nodes into standard CAN systems.

2. Basic Principles of CAN FD

The development of CAN FD was based on the standard CAN protocol and had the requirement to accelerate the serial communication while keeping the physical layer of CAN unchanged. CAN FD started with a similar approach as proposed in [2], increasing the bandwidth by modification of the frame format. Two changes suggest themselves. Firstly, improving the header to payload ratio by allowing longer data fields. Secondly, speeding up the frames by shortening the bit time.

But these steps are only the groundwork, some additional measures are needed, e.g. to keep the Hamming distance of the longer frames at the same level as in standard CAN and to account for the CAN transceiver's loop delay.

The CRC polynomial of CAN is suited for patterns of up to 127 bit in length including the CRC sequence itself. Increasing the CAN frame's payload makes longer polynomials necessary, see section 4.

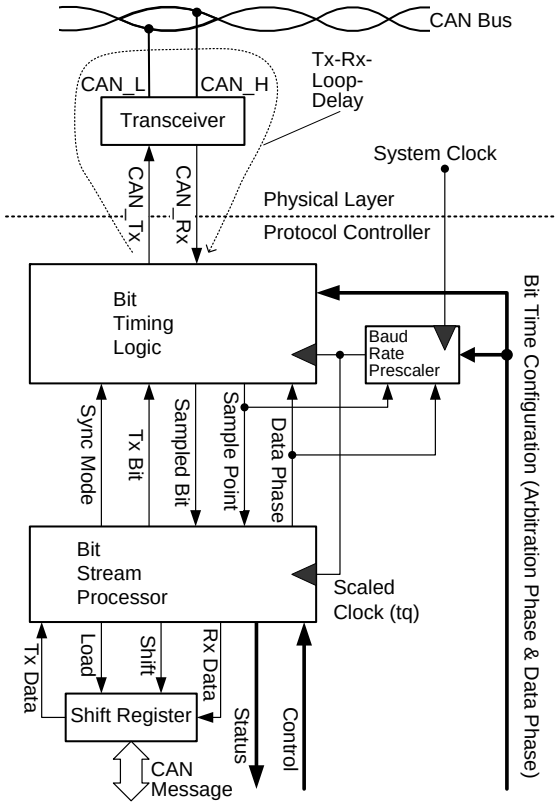

Figure 1 Components of a CAN Node

In a CAN protocol controller, the Bit Timing Logic (BTL) state machine is evaluated once each time quantum and synchronizes the position of the Sample-Point to a specific phase in relation to the edges in the monitored bit stream. Once each CAN bit time, at the Sample-Point, the bit value is decided and the Bit Stream Processor (BSP) state machine is evaluated to decode (in transmitters to encode) the CAN frame. A shift register links the frame's serial bit stream with the controller's message memory, see Figure 1.

CAN nodes synchronize on received edges from recessive to dominant on the CAN bus line. The phases of their Sample-Points are shifted relative to the phase of the transmitter's Sample-Point. A node's specific phase shift depends on the signal delay time from the transmitter to that specific node.

The signal delay time between the nodes needs to be considered when more than one node may transmit a dominant bit. This is the case in the arbitration field or in the acknowledge slot. The configuration of the CAN bit time, especially the Propagation Segment's length and the Sample-Point's position, must ensure that twice the maximum phase shift fits between the Synchronization Segment and the Sample-Point. Once the arbitration is decided, until the end of the CRC Field, only one node transmits dominant bits, all other nodes synchronize themselves to this single transmitter. Therefore it is possible to switch to a predefined (shorter) bit time in this part of a CAN frame, here called the Data-Phase, see Figure 4. The rest of the frame, outside the Data-Phase, is called the Arbitration Phase.

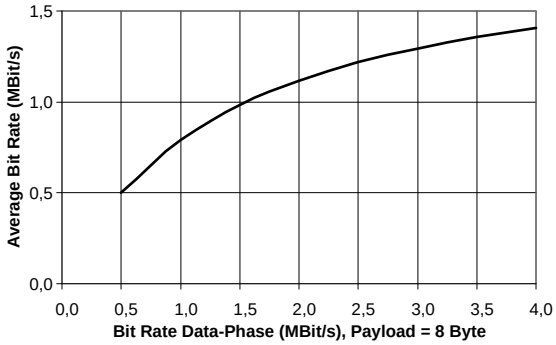

Figure 2 Speeding up from 0.5 MBit/s to 4 MBit/s

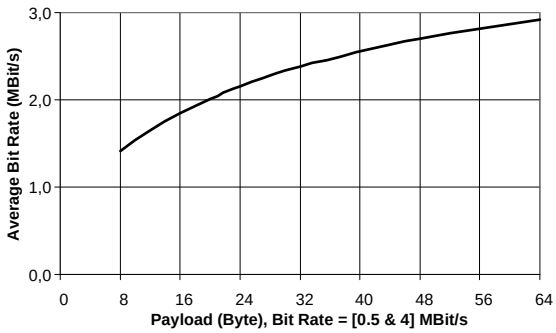

Figure 3 Enlarging a frame to 64 data bytes

All nodes in the network must switch to this shorter bit time synchronously at the start of the Data-Phase and back to the standard bit time at the end of the Data-Phase. Figure 2 shows an example for the average bit rate that can be achieved with a bit rate of 0.5 MBit/s in the Arbitration-Phase and a higher bit rate in the Data-Phase. In the example, an 11-bit identifier is used, there are 8 data bytes in the frame, and stuff bits are not considered.

The example is extended in Figure 3. Here the average bit rate is further increased by lengthening the Data Phase. The bit rate is 0.5 MBit/s in the Arbitration-Phase and 4 MBit/s in the Data-Phase.

The factor between the short bit time in the Data-Phase and the standard bit time in the Arbitration-Phase decides how much the frames are speeded up. This factor has two limits. The first is the speed of the transceivers: bits that are too short cannot be decoded. The second is the time resolution of the CAN synchronization mechanism: after switching to the short bit time, a phase error of one time quantum in the standard bit time needs to be compensated for, see section 5.

At the last bit of the Data-Phase, the CRC Delimiter, all nodes switch back to the standard bit time before the receivers send their acknowledge bit. Receivers are synchronized to the transmitter, but node-specific signal propagation times causes acknowledge bits of the most distant receivers to arrive after that of the nearest receivers. A CAN FD transmitter therefore has to tolerate a two bit long CRC Delimiter before the acknowledge bit. All CAN FD nodes have to tolerate two consecutive dominant bits in the Acknowledge Slot. Latest the second dominant acknowledge bit must be followed by a recessive Acknowledge Delimiter and End of Frame.

CAN's fault confinement strategy, where a node that detects an error in an ongoing frame immediately notifies all other nodes by destroying that frame with an error flag, requires that all nodes monitor their own transmitted bits to check for bit errors. Current CAN transceivers may have, according to ISO 11898-5, a loop delay (CAN_Tx pin to CAN_Rx pin) of up to 255 ns. That means to detect a bit error inside a bit time of the Data-Phase, this bit time has to be significantly longer than the loop delay. To make the length of a short bit time independent of the transceiver's loop delay, CAN FD provides the Transceiver Delay Compensation option, see section 6.

3. CAN FD Frame Format

The Control Field in standard CAN frames contains reserved bits which are specified to be transmitted dominant. In a CAN FD frame, the reserved bit after the **IDE** bit (11-bit Identifier) or after the **RTR** bit (29-bit Identifier) is redefined as Extended Data Length (**EDL**) bit and is transmitted recessive. This sets the receiving BSP and BTL FSMs into CAN FD decoding mode.

The following bits are new in CAN FD :

EDL Extended Data Length
r1, r0 reserved, transmitted dominant
BRS Bit Rate Switch
ESI Error State Indicator

The DLC values from 0000_b to 1000_b still code a Data Field length from 0 to 8 bytes, while the (in standard CAN redundant) DLC values from 1001_b to 1111_b are redefined in CAN FD to code Data Fields with a length of up to 64 byte:

DLC	1001	1010	1011	1100	1101	1110	1111
Byte	12	16	20	24	32	48	64

EDL distinguishes between the standard CAN frame format and the CAN FD frame format.

The value of **BRS** decides whether the bit rate in the Data-Phase is the same as in the Arbitration-Phase (**BRS** dominant) or whether the predefined faster bit rate is used in the Data-Phase (**BRS** recessive).

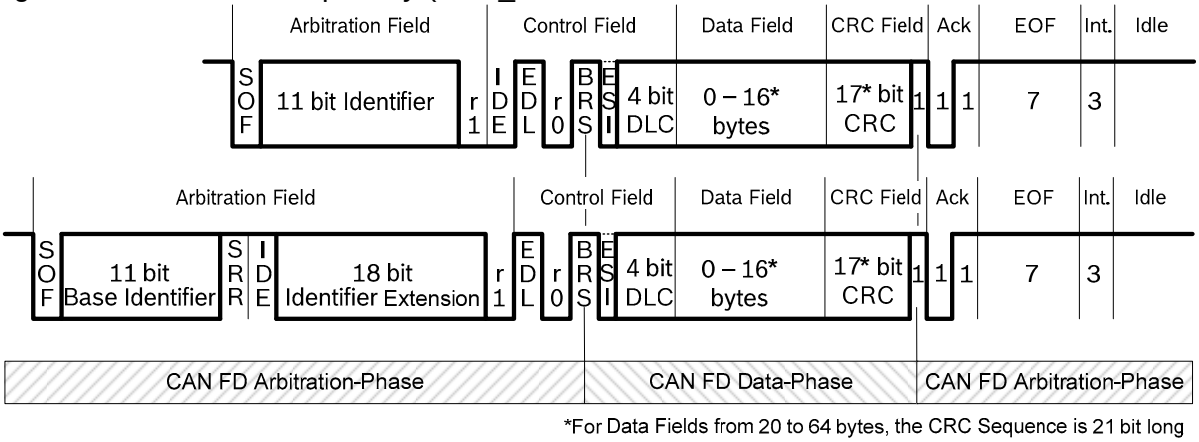

Figure 4 CAN FD Data Frame Format with 11 bit identifier and with 29 bit identifier

In CAN FD frames, **EDL** is always recessive and followed by the dominant bit **r0**. This provides an edge for resynchronization before an optional bit rate switch. The edge is also used to measure the transceiver's loop delay for the optional Transceiver Delay Compensation, see section 6.

In CAN FD frames, the transmitter's error state is indicated by **ESI**, dominant for error active and recessive for error passive. This simplifies network management.

There are no CAN FD remote frames, the bit at the position of the RTR bit in standard CAN frames is replaced by the dominant reserved bit **r1**. However, standard CAN remote frames may optionally be used in CAN FD systems.

Receivers ignore the actual values of the bits **r1** and **r0** in CAN FD frames; they are reserved for future expansion of the protocol, e.g. using **r1** as additional identifier bit.

4. Cyclic Redundancy Check

The error detection capabilities and operational safety of the standard CAN protocol are discussed in [7], [8], and [9]. CAN FD maintains all of CAN's fault confinement mechanisms, including Error Frames, error counters, error-active/ -passive modes, and positive acknowledging for fault-free messages. Since CAN FD allows longer data fields than standard CAN, the CRC Sequence needs to be adapted in order to keep the frame's Hamming Distance at the same value of 6. We chose two new BCH-type CRC polynomials: g17 for frames with up to 16 data bytes, g21 for frames with more than 16 data bytes.

$$g17 = x^{17}+x^{16}+x^{14}+x^{13}+x^{11}+x^{6}+x^{4}+x^{3}+x^{1}+1$$

$$g21 = x^{21}+x^{20}+x^{13}+x^{11}+x^{7}+x^{4}+x^{3}+1$$

For this reason, the length of the CRC Sequence in CAN FD data frames depends on the Data Length Code DLC. At the beginning of a frame, all nodes, including the transmitter, start to calculate the frame's CRC Sequence according to all three polynomials, g17, g21, and the standard CAN polynomial. When the frame format is decided in the Control Field and the DLC is transmitted, one of the three polynomials is selected. The transmitter uses the selected polynomial to generate the frame's CRC Sequence. The receivers use the applicable polynomial to decide whether the frame is to be acknowledged.

In standard CAN, the stuff bits, which are inserted into the bit stream to ensure that there are enough edges for resynchronization, are not considered for CRC calculation. As described e.g. in [7], two bit errors may on rare occasion remain undetected when the first generates a bit stuffing condition and the second then removes a stuff condition (or vice versa), shifting the position of the frame bits between the two bit errors. The shifted area may lead to a burst error that is too long for the CRC mechanism.

The treatment of stuff bits in CAN FD is changed to ensure that this cannot happen. The simplest measure would have been to include all stuff bits into the CRC calculation. However, this would prevent the well proven CRC hardware implementation with the feedback shift register that calculates the CRC Sequence while the frame is in progress. Our solution consists of two measures: Including the stuff bits preceding the CRC Sequence into the CRC calculation and changing the stuffing mechanism for the CRC Sequence. Contrary to the standard CAN stuffing method, where a stuff bit of inverse polarity is inserted after every five consecutive bits of the same polarity, the positions of the stuff bits in the CAN FD's CRC Sequence are fixed: The CRC Sequence starts with a stuff bit and additional stuff bits are inserted after every four bits of the sequence. Each of these fixed stuff bits has the inverse polarity of its preceding bit. The number of stuff bits in the CRC Sequence is equal to the maximum number of stuff bits according to the standard CAN stuffing mechanism. As in the standard CAN stuffing mechanism, the maximum number of consecutive bits with the same value is five, the maximum distance between edges for resynchronization is ten.

5. CAN Bit Time Switching

There are two sets of configuration registers in CAN FD: The first for the standard bit time in the Arbitration-Phase and the second for the bit time in the Data-Phase.

The BTL and Baud Rate Prescaler (BRP) FSMs (see Figure 1) switch to the second bit time configuration at the Sample-Point where the **BRS** bit is sampled recessive. They switch back to the first bit time at the Sample-Point of the CRC Delimiter, or

when an error condition is detected that causes an error frame.

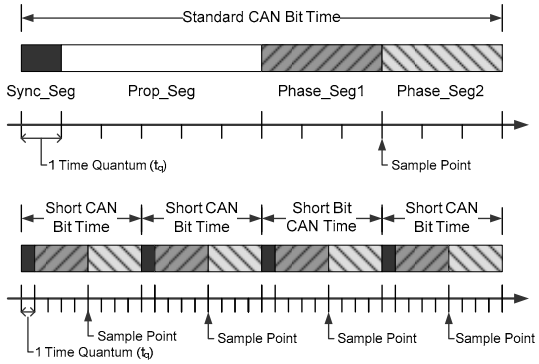

Figure 5 Standard and Short CAN Bit Time

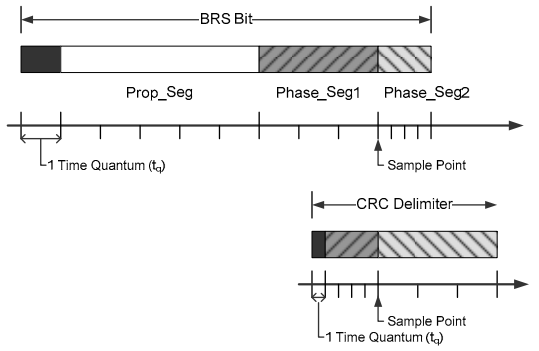

Figure 6 Bit Time at **BRS** Bit and CRC Delimiter

Figure 5 shows an example for the bit time configurations where the data rate in the Data-Phase is four times faster than in the Arbitration-Phase. Both, the length of the t_q and the number of t_q in the bit time may be different in the two configurations. The two configurations may be identical, but the bit time in the Data-Phase may not be longer than in the Arbitration-Phase. The two bits where the switch happens are of intermediate length, since the configurations are switched at Sample-Points (see Figure 6). Together the two bits are as long as the sum of one of each of the bit times.

Switching the bit time configurations at the Sample-Point instead of after the end of Phase_Seg2 is necessary to ensure that a following synchronization is performed in all nodes according to the parameters of the second bit time configuration. Phase-shifts between the nodes may result in not all of them agreeing on the border between Phase_Seg2 and the subsequent Sync_Seg.

Figure 7 shows the simulation of a test case where CAN_0 and CAN_1 arbitrate for the CAN bus. The signals can_tx and can_rx are the interface between the protocol controllers and the transceivers. The sample_point shows where the can_rx input is captured. The signals f_tx and f_rx show where the bit rate is switched, they could be used for mode-switching in new - CAN FD optimized - transceivers, enabling even higher bit rates in the Data-Phase.

Both nodes send the same base identifier. CAN_0 sends a CAN FD Frame with 11-bit identifier while CAN_1 sends an extended frame and loses arbitration at the **SRR** bit.

Transmitters do not synchronize on “late” edges (those detected between Sync_Seg and Sample-Point), otherwise the transceiver loop delay would cause them to lengthen dominant bits. So as transmitter, CAN_1 did not synchronize on CAN_0 before the edge from EDL to **r0**.

In the simulated test case, there is a delay of 433 ns between the nodes; they use a bit rate of 1 MBit/s in the Arbitration-Phase and 10 MBit/s in the Data-Phase. At the **SRR** bit, where CAN_1 loses arbitration, its Sample-Point is 350 ns (see strobes 1 and 2) earlier than that of CAN_0.

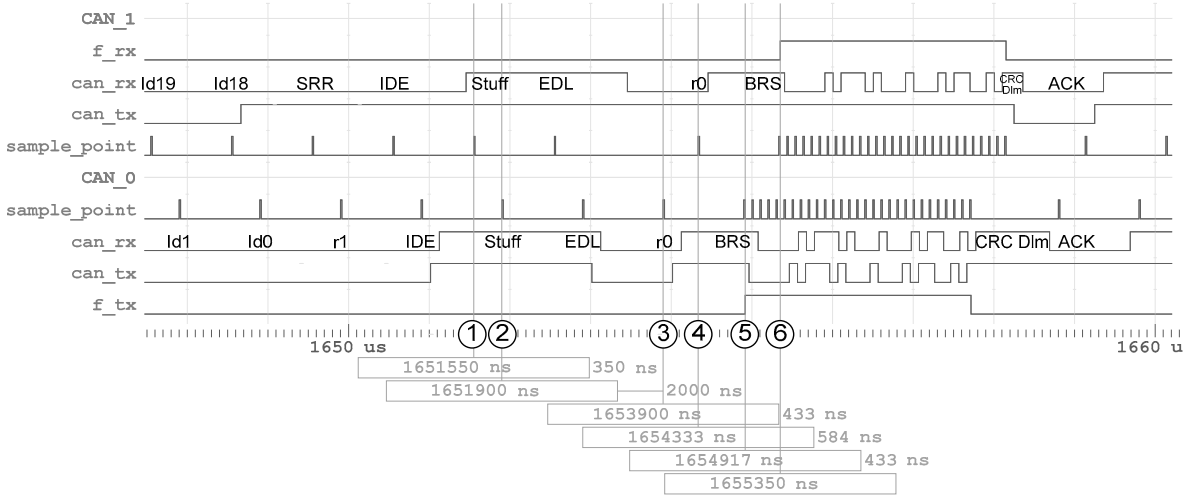

Figure 7 CAN FD Bit Time Switching after Arbitration

CAN_1 synchronizes to CAN_0 at the edge from **EDL** to **r0**. Afterwards its Sample-Point comes 433 ns (the signal propagation time between the nodes) after that of CAN_0 (see strobes 3 and 4). Both nodes switch their bit rate at the Sample-Points of their **BRS** bits (see strobes 5 and 6). The signal f_tx shows the transmitter's Data-Phase, f_rx the receiver's. They both are reset at the CRC Delimiter, before the Acknowledge bit sent by CAN_1.

The CRC Delimiter seen by the transmitter CAN_0 is prolonged by the signal propagation time, the Acknowledge bit conforms to the Arbitration-Phase's bit rate.

The analog input signal at CAN_Rx needs to be synchronized to the clock of the BTL FSM. Together with the BTL's time step size of 1 t_q, this digitization delay limits the time resolution of the CAN bit synchronization. This means a phase error of up to 1 t_q may remain after a (re-)synchronization; the synchronization quality depends on the duration of the time quantum. The Sync_Seg with a fixed duration of one t_q compensates for this residual phase error in CAN bit timing, but one t_q in the first bit time may correspond to several t_q in the second bit time. The maximum possible residual phase error has to be taken into account for the configuration. Setting t_q to the same duration in both configurations maximizes the tolerance range.

In existing CAN implementations, the maximum number of time quanta in a bit time is 25, while the duration of the time quantum is defined by the controller's clock period and the BRP. This allows only few combinations of bit time configurations for the Arbitration-Phase and for the Data-Phase with the same t_q duration.

In automotive applications, with a bit rate of e.g. 0.5 MBit/s or 1 MBit/s in the Arbitration-Phase, the acceleration in the Data-Phase is limited to a factor of about 5. The reasons for this limit are the minimum pulse width in the receive path of currently available transceivers and EMI considerations. In other applications, long bus lines may limit the bit rate in the Arbitration-Phase to e.g. 125 KBit/s, enabling a higher acceleration factor.

Figure 8 shows how the average bit rate of a CAN network that needs a bit time of 8 µs in Arbitration-Phase can be accelerated without exceeding the specification range of existing CAN transceivers in the Data-Phase. Figure 9 shows how this acceleration is increased when the data field gets longer. The advantage of the improved header to payload ratio rises with the acceleration factor between Arbitration-Phase and Data-Phase.

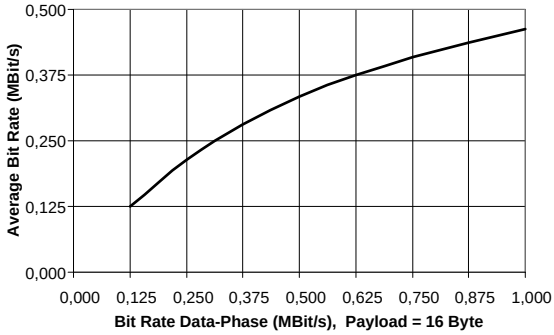

Figure 8 CAN FD Example for long Bus Lines

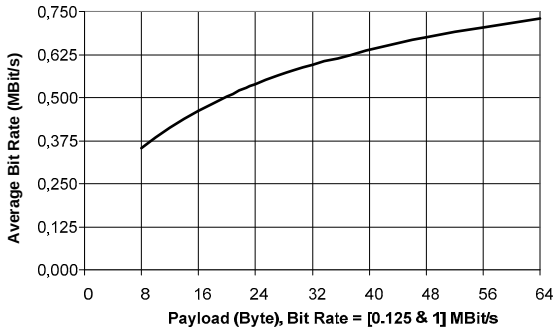

Figure 9 Average Bit Rates for long Bus Lines

ISO 11898-1 allows more than 8 t_q for each of the bit time segments Prop_Seg, Phase_Seg1, and Phase_Seg2. We increased the configuration range to 16 t_q for Phase_Seg2 and to 64 t_q for the sum of Prop_Seg and Phase_Seg1 in our CAN FD implementation. This allows a wide range of bit time combinations with the same time quantum length. The range of the SJW configuration is also increased to 16 t_q for CAN FD applications. This enables a high acceleration factor with a low residual phase error at the **BRS** bit.

6. Transceiver Delay Compensation

Current CAN transceivers may have, according to ISO 11898-5, a loop delay (from the CAN_Tx pin to the CAN_Rx pin) of up to 255 ns. Since transmitters are required to check for errors in their transmitted bits, this would set a lower limit for the bit time in the Data-Phase if the check needs to be done at the bit's Sample-Point.

Measurements have shown that existing CAN transceivers are able to transmit and receive bits that are shorter than their loop delay. In this case the check for bit errors

needs to be delayed until the bit value which is transmitted at the CAN_Tx output is looped back to the CAN_Rx input. This is the purpose of the optional CAN FD Transceiver Delay Compensation mechanism. Receivers do not need this mechanism. Transmitters apply it in the Data-Phase of a frame.

The point in time where the looped back bit value is checked is named the secondary sample point (SSP). The actual loop delay is not a static value, it depends - apart from silicon parameters - mainly on the operating temperature.

The CAN FD protocol controller is able to perform a delay measurement to find the optimum position for the SSP. Within each CAN FD frame, the transmitter measures the delay between the data transmitted at the CAN_Tx output and the data received at the CAN_Rx input. The measurement is performed when the arbitration is decided but before the bit rate is switched, at the edge from **EDL** bit to **r0** bit. The delay is measured (in system clock periods) by a counter that starts at the beginning of the bit **r0** at CAN_Tx and stops when the edge is seen at CAN_Rx, see Figure 10.

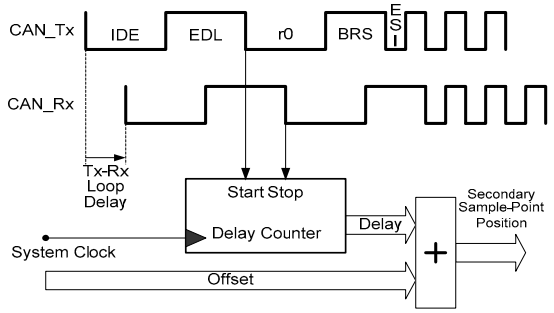

Figure 10 Measuring the Transceiver Loop Delay

The result is a node-specific value. It does not depend on signal propagation times on the CAN bus line. A configurable offset is added to the measured delay value to place the SSP into the middle of the bits seen at CAN_Rx

When the Transceiver Delay Compensation mechanism is enabled, it changes the way how a transmitter checks for bit errors during the Data-Phase of a CAN FD frame from direct comparison of transmitted and received bits at the CAN Sample-Point to a delayed comparison at the SSP.

The position of the SSP is always relative to the start of a transmitted bit. It may be more than one bit time after the end of that bit. Transmitted bits are buffered until the SSP is reached. Then their value is compared with the actual value of the input signal to check for bit errors. If a bit error is detected, this information is buffered until the next CAN Sample-Point is reached where it is presented to the BSP FSM. The BSP FSM answers to the bit error according to the rules of the CAN fault confinement with an error frame; the bit rate is switched back to that of the Arbitration-Phase. When no bit error is detected until the Sample-Point of the CRC Delimiter is reached, the CAN FD protocol controller switches back the bit rate and returns to standard bit error checking. The actual value of the CRC Delimiter bit is disregarded by transmitters using the Transceiver Delay Compensation mechanism. A global error at the end of the CRC Field will cause the receivers to send error frames that the transmitter will detect during Acknowledge or End of Frame.

7. Measurements

The development of the CAN FD protocol went in parallel with the design of CAN FD protocol controllers for simulative verification and for laboratory evaluation. Main topics of the analysis were the new protocol features and the limits set by the physical layer. The measurements were based on FPGA implementations of CAN FD and a multi-node CAN network with standard CAN transceivers (e.g. NXP TJA1040). For the measurements shown here, the network consists of 7 nodes connected by a linear CAN bus line. The distance between the terminations at node T2 and node R9 is 42 m, the bit rate switches from an Arbitration-Phase at 0.5 MBit/s to Data-Phases at 15 MBit/s or at 12 MBit/s.

It is not expected a bit rate of 15 MBit/s can be reached in automotive conditions with existing transceivers. The examples intend to show that the bit rate in the Data-Phase is not limited by the signal propagation time in the transceivers and on the CAN bus line.

Special attention was given to the effects of the transceiver loop delay, see Figure 11. In this example, the transceiver's loop delay is 126 ns at room temperature. This is almost twice the Data-Phase's bit time, here 66.67 ns @15 MBit/s. The output pin T2_Tx already starts the DLC before the **ESI** bit reaches the input pin T2_Rx or, af-

ter the CAN bus delays, the receivers' input pins R3_Rx and R9_Rx.

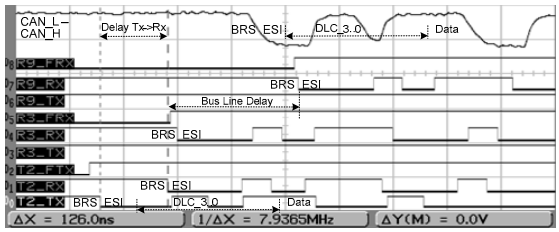

Figure 11 CAN FD Transceiver Loop Delay

The example in Figure 12 shows a complete CAN FD frame with 29 bit Identifier and 64 data bytes. Here the complete Data-Phase is (at 12 MBit/s) shorter than 23 bits of the Arbitration-Phase. In the example, there is CAN arbitration in the first bits of the identifier, superposition of Acknowledge bits from near and from distant receivers virtually prolongs that bit.

Conclusion

CAN FD is a new protocol that combines CAN's core features with a higher data rate. For automotive applications, CAN FD targets an average data rate of 2.5 MBit/s with existing CAN transceivers, resulting in the same effective payload as a low-speed FlexRay network. There is an easy migration path from CAN systems to CAN FD systems since CAN application software can be left (apart from configuration) unchanged. CAN FD controllers can operate in CAN systems. The Bosch CAN IP modules are currently being adapted to optionally support the CAN FD protocol.

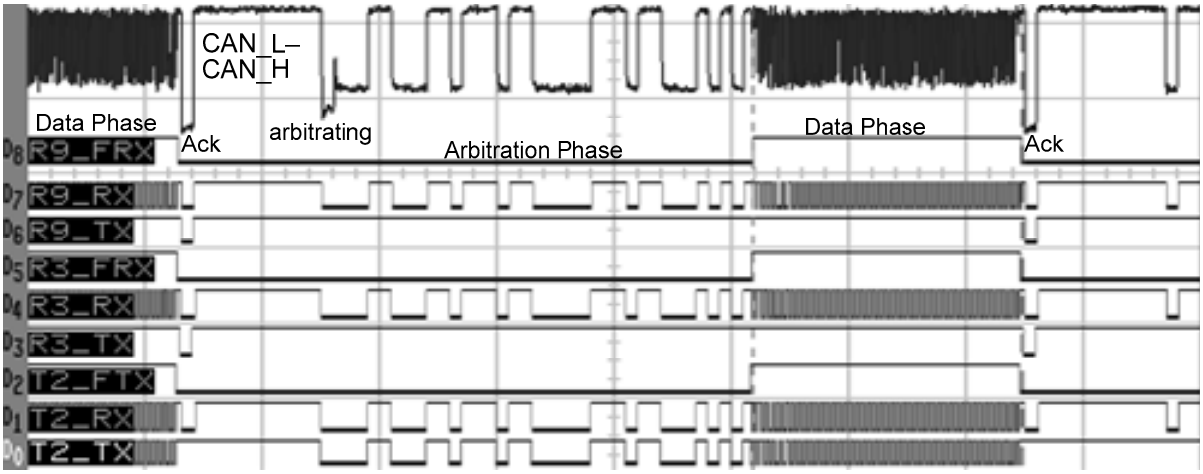

Figure 12 CAN FD Frame with 64 Data Bytes and 12 MBit/s in the Data-Phase

8. References

[1] Robert Bosch GmbH; CAN with Flexible Data-Rate; Version 1.1; Date Aug.2011, http://www.semiconductors.bosch.de/media/... pdf/canliteratur/can_fd.pdf

[2] G. Cena, A. Valenzano; Overclocking of Controller Area Networks; IEE Electronics Letters, vol. 35, 1999, pp. 1923–1925, No. 22, 28th October.

[3] G. Cena, L. Durante, A. Valenzano; A new CAN-like field network based on a star topology; Computer Standards &Interfaces, vol. 23, pp. 209–222, July 2001.

[4] I. Sheikh and M. Short; Improving information throughput in CAN networks: Implementing the dual-speed approach; in 8th International Workshop on Real-Time Networks RTN'09, Dublin, Ireland, June 2009.

[5] R. Kurachi, M. Nishimura et al.; Development of Scalable CAN Protocol; SEI Technical Review No 71, October 2010, pp. 31-36.

[6] S. Ziermann, T. Wildermann, J. Teich; CAN+: A new backward compatible Controller Area Network (CAN) protocol with up to 16x higher data rates; Proceedings of DATE 2009, IEEE Computer Society, Nice, France, pp. 1088-1093, April 20-24, 2009.

[7] J. Unruh, H.-J. Mathony, K.-H. Kaiser; Error Detection Analysis of Automotive Communication Protocols; SAE Paper 900699, Detroit, USA, 1990

[8] J. Charzinski; Performance of the Error Detection Mechanisms in CAN; Proceedings of the 1st international CAN Conference, Mainz, Germany, September 1994, pp. 1.20-1.29.

[9] E. Tran; Multi-Bit Error Vulnerabilities in the Controller Area Network Protocol; Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pa, 1999.

[10] R.Obermaisser, R. Kammerer, A. Kasper; Sternkoppler für Controller Area Network (CAN) auf Basis eines Multi-Processor System-on-a-Chip (MPSoC); Proceedings of the AmE 2011, Dortmund, Germany, May 04, 2011

Author	Florian Hartwich
Company	Robert Bosch GmbH
Address	Postfach 13 42
	72703 Reutlingen, Germany
Phone	+49 07121 35-2594
Fax	+49 0711 811-5142594
E-mail	Florian.Hartwich¤de.bosch.com
Website	http://www.can.bosch.com