CAN with Flexible Data-Rate

White Paper
Version 1.1

Recital

The acceptance and introduction of serial communication to more and more applications has led to requirements that the bandwidth for the serial communication needs to be increased.

Two factors limit the effective data-rate in CAN networks, first the minimum bit length required for the function of the CAN bus arbitration method and second the relation between the numbers of data bits and other frame bits in a CAN message.

This white paper describes a new protocol that is called "CAN with Flexible Data-Rate" or CAN FD and is based on the CAN protocol as specified in ISO 11898-1. It still uses the CAN bus arbitration method, it increases the bit-rate by switching to a shorter bit time after the end of the arbitration process and returns to the longer bit time at the CRC Delimiter, before the receivers send their acknowledge bits. The effective data-rate is increased by allowing longer data fields. CAN uses four bits as Data Length Code resulting in 16 different codes, but only the first nine values are used, codes [0 - 8] standing for a data field length of [0 - 8] bytes. In CAN, the codes [9 - 15] are defined to signify eight data bytes. In CAN FD, the codes are used to signify longer data fields.

Note
CAN systems can migrate gradually to CAN FD systems. All nodes in the network must have a CAN FD protocol controller for CAN FD communication, but all CAN FD protocol controllers are also able to take part in standard CAN communication. In the introductory phase, CAN FD communication may be limited to specific use cases, e.g. software-download, while the other nodes that do not support CAN FD are kept in standby. If the CAN FD communication is limited to data fields with a length of up to eight data bytes, it is not necessary to change the application program apart from the initial configuration of the controller.

Content

1   INTRODUCTION

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed realtime control with a very high level of security.

Its domain of application ranges from high speed networks to low cost multiplex wiring. In automotive electronics, engine control units, sensors, anti-skid-systems, etc. are connected using CAN with bitrates up to 1 Mbit/s. At the same time it is cost effective to build into vehicle body electronics, e.g. lamp clusters, electric windows etc. to replace the wiring harness otherwise required.

CAN FD (CAN with Flexible Data-Rate) complements CAN in applications that require a higher data-rate. CAN FD protocol controllers are also able to take part in standard CAN communication, making it possible to use CAN FD only in specific operation modes, e.g. software-download at end-of-line programming or at maintenance, while other controllers that do not support CAN FD are kept in standby.

CAN FD makes use of the (first) reserved bit of a CAN frame. In a CAN frame, this reserved bit is transmitted 'dominant'. In a CAN FD frame, this bit is transmitted 'recessive' and indicates that this frame is transmitted in the CAN FD format. A CAN FD controller is able to decode both formats.

There are two main differences between the CAN FD frame format and the CAN frame format, first the option to use frames with more than 8 data bytes and second the option to switch to a different bit rate after the arbitration is decided.

CAN FD requires two sets of bit timing configuration registers, the first bit time for the arbitration phase and the second bit time for the data phase. The bit time for the arbitration phase has the same limitations as in standard CAN networks, the bit time for the data phase is chosen with regard to the performance of the chosen transceiver and the characteristics of the CAN FD network. Both bit times may be identical.

Standard CAN transceivers can be used for CAN FD, dedicated transceivers are optional. CAN FD protocol controllers may provide additional interface signals to switch - in the phase with higher bit-rate - a dedicated CAN FD transceiver into an alternate operating mode.

Dedicated CAN FD transceivers may use an alternate coding system in the phase with higher bit-rate, not restricted to CAN's NRZ coding.

2 BASIC CONCEPTS

A CAN FD frame consists of the same elements as a CAN frame, the difference is that in a CAN FD frame, the Data Field and the CRC Field may be longer. Message validation requires, as in CAN, a dominant Acknowledge bit from at least one receiver. The CAN FD fault confinement with Error Frames, Error Counters, Error Passive level and Bus-Off level is the same as in CAN, it is based on the same five error types: Bit Error, Stuff Error, CRC Error, Form Error, and Acknowledgement Error.
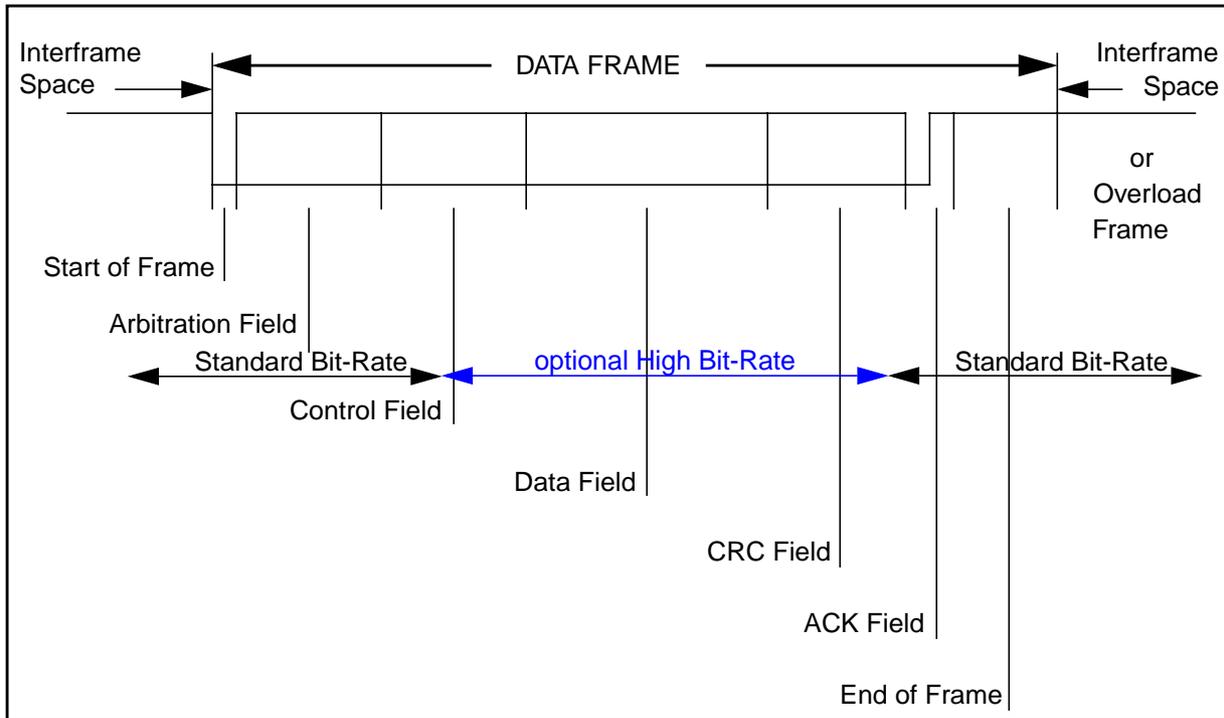
CAN FD frames have the same structure as CAN frames, the differentiation between CAN frames and CAN FD frames is at the reserved bit immediately after the end of arbitration, before the Data Length Code in the Control Field. In a CAN FD frame, this reserved bit is transmitted 'recessive' and is renamed EDL (Extended Data Length).
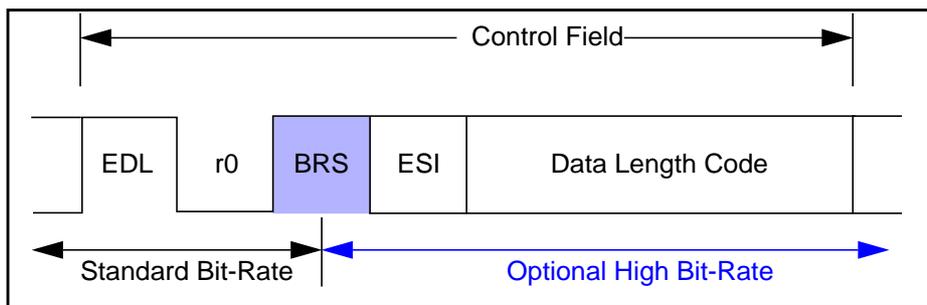

In frames with 11-bit identifiers, EDL comes after the IDE bit, in frames with 29-bit-identifier, it comes after the RTR bit. EDL is always followed by the 'dominant' bit r0, which is reserved for future expansion of the protocol.

The Control Field of the CAN FD frame contains two additional bits, BRS and ESI. BRS (Bit Rate Switch) decides whether the bit rate is switched inside this frame. If it is transmitted 'recessive', the bit rate is switched from the standard bit rate to the preconfigured alternate bit rate. If it transmitted 'dominant', the bit rate is not switched.

The bit ESI (Error State Indicator) is transmitted 'dominant' by a node that is error-active, 'recessive' by a node that is error-passive.

The first part of a CAN FD frame, until the BRS bit, is transmitted with the same bit-rate as a CAN frame. The bit-rate is switched if the BRS bit is 'recessive', until the CRC Delimiter is reached or until the CAN FD controller sees an error condition that results in the starting of an Error Frame. CAN FD Error Frames, as well as ACK Field, End of Frame, and Overload Frames are transmitted with the same bit-rate as a CAN Error Frame

3 FRAME FORMAT

Standard Format
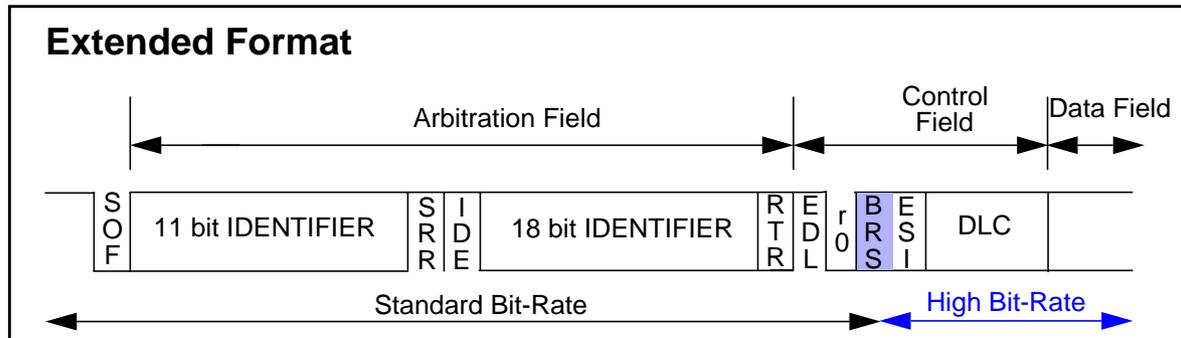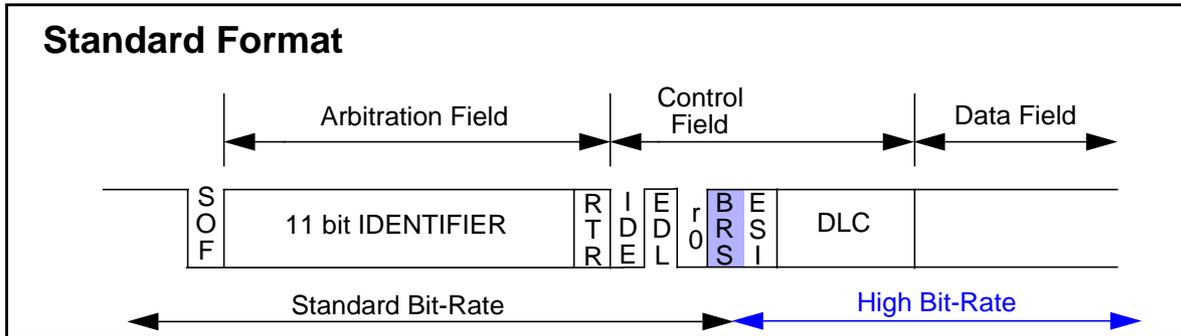
Extended Format

CAN FD supports both Identifier lengths of the CAN protocol, the 11 bit long "Standard Format" and the 29 bit long "Extended Format". In both cases, the bit-rate is switched to the shorter bit time at the BRS bit (before the ESI bit and the DLC).

DATA LENGTH CODE
The number of bytes in the DATA FIELD is indicated by the DATA LENGTH CODE. This DATA LENGTH CODE is 4 bits wide and is transmitted within the CONTROL FIELD.

The coding of the DATA LENGTH CODE is different in CAN FD. The first nine codes are the same, but the following codes, that in CAN specify a DATA FIELD of eight bytes, specify longer DATA FIELDS in CAN FD. All Remote Frames shall use a DATA LENGTH CODE of zero, regardless of the DATA LENGTH CODE of the corresponding Data Frame.

Note
In this document, the maximum length of the DATA FIELD is specified to be 64 bytes. This value, and the other values > 8 of DATA LENGTH CODE may change in the final specification of CAN FD.

Coding of the number of data bytes by the DATA LENGTH CODE:

	Number of Data Bytes	Data Length Cod			
		DLC3	DLC2	DLC1	DLC0
ISO 11898-1	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	0	1	1
	4	0	1	0	0
	5	0	1	0	1
	6	0	1	1	0
	7	0	1	1	1
	8	1	0	0	0
Proposal for Additional Codes in CAN FD	12	1	0	0	1
	16	1	0	1	0
	20	1	0	1	1
	24	1	1	0	0
	32	1	1	0	1
	48	1	1	1	0
	64	1	1	1	1

CRC FIELD
contains the CRC SEQUENCE followed by a CRC DELIMITER
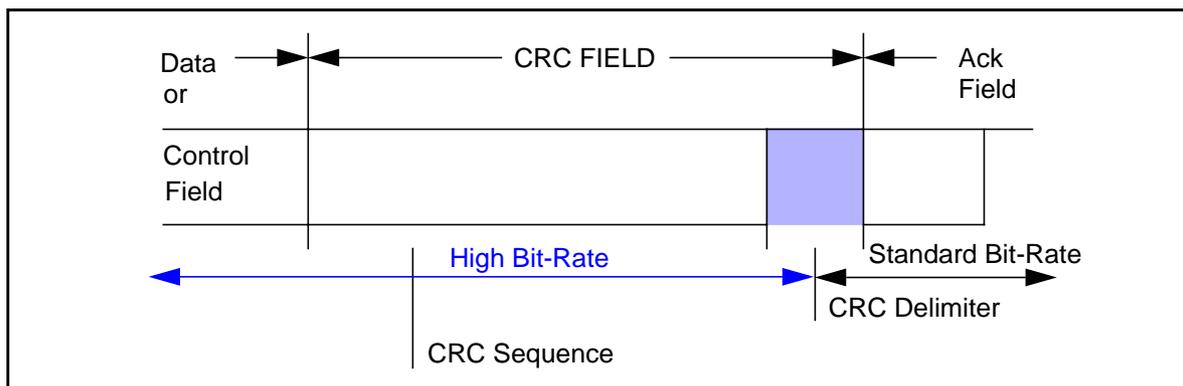
CRC SEQUENCE
The frame check sequence is derived from a cyclic redundancy code (BCH Code).
In order to carry out the CRC calculation, the polynomial to be divided is defined by the coefficients of the relevant bit stream. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial.

The relevant bit stream for CRC calculation is the bit stream, including stuff bits, consisting of START OF FRAME, ARBITRATION FIELD, CONTROL FIELD, DATA FIELD (if present) and, for the n (n is the order of the generator-polynomial) lowest coefficients, by 0.

CAN FD uses different CRC polynomials for different frame length. The first polynomial, CRC_17, is used for frames with up to sixteen data bytes, the second polynomial, is used for frames with more than sixteen data bytes.

CRC_17, with a Hamming Distance HD = 6, is best suited for frames with bit counts less than 210 bits:

$$X^{17} + X^{16} + X^{14} + X^{13} + X^{11} + X^6 + X^4 + X^3 + X^1 + 1.$$

For frames with more than sixteen bytes in the DATA FIELD, a different (and longer) CRC polynomial is used, adapted to the maximum length of the frame. The CRC Field is lengthened accordingly.

Each CRC SEQUENCE is calculated in a separate shift register block. At the start of the frame, in all nodes all CRC SEQUENCES shall be calculated concurrently, until after the arbitration one of the CRC SEQUENCES is selected by the EDL bit and by the DLC. Only the selected CRC SEQUENCE can cause a CRC Error.

In CAN FD frames, the stuff bits that occur before the CRC SEQUENCE shall be protected by the CRC. The stuff bits that may occur in the CRC SEQUENCE shall not be taken into account for CRC calculation or CRC check. The CAN bit stuffing method is changed for the CRC SEQUENCE, the stuff bits shall be inserted at fixed positions. There shall be a fixed stuff bit before the first bit of the CRC SEQUENCE, even if the last bits of the preceding field do not fulfill the CAN stuff condition. A further stuff bit shall be inserted after each fourth bit of the CRC SEQUENCE. The value of such a fixed stuff bit shall be the inverse value of the bit preceding the fixed stuff bit. A receiver shall discard the fixed stuff bits from the bit stream for the CRC check, it shall detect a stuff error if the fixed stuff bit has the same value as its preceding bit.

Note
The actual CRC polynomials will be defined after the coding of the DATA LENGTH CODE is finalized.

CRC DELIMITER
The CRC SEQUENCE is followed by the CRC DELIMITER which consists of one or two 'recessive' bits. A transmitter shall send only one 'recessive' bit as CRC DELIMITER but all transmitting nodes shall accept two 'recessive' bits before the edge from recessive to dominant that starts the Acknowledge Slot. A receiver will send its Acknowledge bit after the first CRC DELIMITER bit.

Note
When the CRC Delimiter is detected, the CAN FD protocol controllers switch back to the bit-rate with the longer bit time.

The phase-shift between the nodes in a CAN network is defined by the delay times in

the transceivers and the propagation time on the CAN bus line. The phase-shift is the same in CAN and in CAN FD, but it is proportionally larger in the phase with the shorter bit time. All receivers in the network may have a different phase-shift to the transmitter, depending on their distances from the transmitter, since they see the transmitted edges at different times. To compensate for these phase-shifts when the bit-rate is switched back to the longer bit time, one additional bit time tolerance is allowed before and after the edge from recessive to dominant that starts the Acknowledge Slot.

ACK FIELD
The ACK FIELD is two or three bits long and contains the ACK SLOT and the ACK DELIMITER. In the ACK FIELD the transmitting station sends two 'recessive' bits.

A RECEIVER which has received a valid message correctly, reports this to the TRANSMITTER by sending one 'dominant' bit at the start of the ACK SLOT.
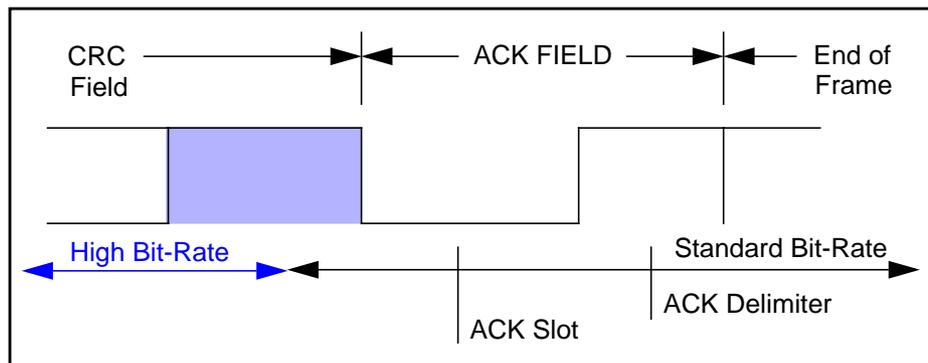
ACK SLOT
All stations having received the matching CRC SEQUENCE report this within the ACK SLOT by superscribing the 'recessive' bit of the TRANSMITTER with one 'dominant' bit (they send 'ACK'). To compensate for phase shifts between the receivers, all nodes shall accept a two bit long 'dominant' phase of overlapping ACK bits as a valid ACK.

ACK DELIMITER
The ACK DELIMITER is the second (one bit long ACK) or third (two bit long ACK) bit of the ACK FIELD and has to be a 'recessive' bit. As a consequence, the ACK SLOT is surrounded by two 'recessive' bits (CRC DELIMITER, ACK DELIMITER).

END OF FRAME
Each DATA FRAME and REMOTE FRAME is delimited by a flag sequence consisting of seven 'recessive' bits.

4 CAN PROTOCOL FEATURES IN CAN FD

The following parts of the CAN protocol specification also apply unchanged in the CAN FD protocol:

4.1 Definition of TRANSMITTER / RECEIVER

4.2 MESSAGE FILTERING

4.3 MESSAGE VALIDATION

4.4 CODING

4.5 ERROR HANDLING

4.6 Error Detection

4.7 Error Signalling

4.8 FAULT CONFINEMENT

5 BIT TIMING REQUIREMENTS

The CAN FD protocol defines two bit-rates, the first for the arbitration phase with a longer bit time and the second for the data phase with a shorter bit time. The definition for the first bit-rate is the same as for the NOMINAL BIT RATE and the NOMINAL BIT TIME in the CAN protocol specification. The definition for the second bit-rate requires a separate configuration register set. Both bit times consist of separate non-overlapping time segments, these segments form the bit time as shown in this figure:
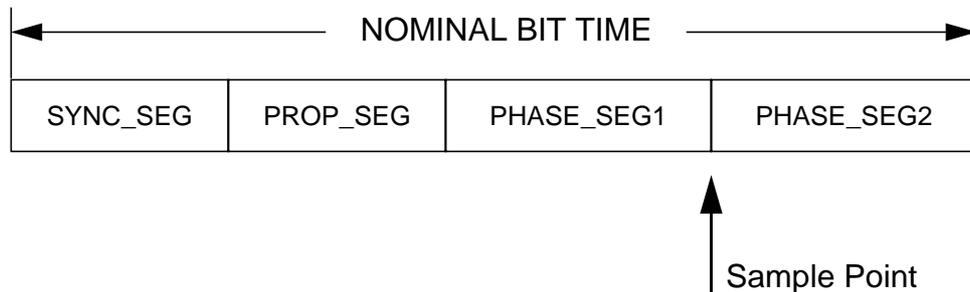
- SYNCHRONIZATION SEGMENT (SYNC_SEG)

This part of the bit time is used to synchronize the various nodes on the bus. An edge is expected to lie within this segment.

- PROPAGATION TIME SEGMENT (PROP_SEG)

This part of the bit time is used to compensate for the physical delay times within the network. It is twice the sum of the signal's propagation time on the bus line, the input comparator delay, and the output driver delay.

- PHASE BUFFER SEGMENT1 (PHASE_SEG1)

- PHASE BUFFER SEGMENT2 (PHASE_SEG2)

These Phase-Buffer-Segments are used to compensate for edge phase errors. These segments can be lengthened or shortened by resynchronization.

SAMPLE POINT

The SAMPLE POINT is the point of time at which the bus level is read and interpreted as the value of that respective bit. It's location is at the end of PHASE_SEG1.

INFORMATION PROCESSING TIME

The INFORMATION PROCESSING TIME is the time segment starting with the SAMPLE POINT reserved for calculation the subsequent bit level.

The length of the time segments is defined in integer multiples of the TIME QUANTUM, with the TIME QUANTUM is a fixed unit of time derived from the oscillator period. There exists a programmable prescaler, with integral values, ranging at least from 1 to 32. Starting with the MINIMUM TIME QUANTUM, the TIME QUANTUM can have a length of

$$\text{TIME QUANTUM}(n) = m(n) * \text{MINIMUM TIME QUANTUM}$$

with m the value of the prescaler. Two values for the prescaler, $m(1)$ and $m(2)$ are defined for the CAN FD protocol, one for each bit-rate, resulting in two different lengths of the TIME QUANTUM.

The time segments for the two bit rates of the CAN FD protocol are defined by two sets of configuration registers.

Length of Time Segments for the first bit-rate

- SYNC_SEG(1) is 1 TIME QUANTUM(1) long.
- PROP_SEG(1) is programmable to be 1,2,...,8 or more TIME QUANTA(1) long.
- PHASE_SEG1(1) is programmable to be 1,2,...,8 or more TIME QUANTA(1) long.
- PHASE_SEG2(1) is the maximum of PHASE_SEG1(1) and the INFORMATION PROCESSING TIME
- The INFORMATION PROCESSING TIME is less than or equal to 2 TIME QUANTA(1) long.

Length of Time Segments for the second bit-rate

- SYNC_SEG(2) is 1 TIME QUANTUM(2) long.
- PROP_SEG(2) is programmable to be 0,1,2,...,8 TIME QUANTA(2) long.
- PHASE_SEG1(2) is programmable to be 1,2,...,8 TIME QUANTA(2) long.
- PHASE_SEG2(2) is the maximum of PHASE_SEG1(2) and the INFORMATION PROCESSING TIME
- The INFORMATION PROCESSING TIME is less than or equal to 2 TIME QUANTA long.

The number of TIME QUANTA in a bit time shall be programmable at least from 8 to 25.

The position of the SAMPLE POINT may differ in the two bit timing configurations, the length of the PROP_SEG may be reduced in the configuration for the second bit-rate. When the bit rate is switched at the BRS bit or at the CRC DELIMITER bit, it shall be switched immediately after the sample point, causing the length of these two bits to be intermediate. The sum of the length of these two bits shall be the same as the sum of one bit of the first bit rate and one bit of the second bit rate. When the bit rate is switched because an error condition is detected, the switching time is shifted after the sample point, by the length of the INFORMATION PROCESSING TIME.

A transmitter shall not resynchronize itself while it transmits in the CAN FD data phase.

5.1 TRANSCEIVER DELAY COMPENSATION

Without a transceiver delay compensation, the bit rate in the fast phase of a CAN FD frame is limited by the fact that the transmitter detects a bit error if it cannot receive its own transmitted bit latest at the SAMPLE POINT of that bit.

CAN FD nodes shall support an optional transceiver delay compensation mechanism, to be used in applications where the length of the CAN bit time in the fast phase is shorter than the limit required by the transceiver's internal delay time.

The transceiver delay compensation mechanism defines a secondary sample point where the transmitted bit is compared with the received bit in order to check for bit errors. This mechanism shall only be used by transmitters in the fast phase of a CAN FD frame, when it is used, the transmitter shall ignore the bit value received at the SAMPLE POINT. The delay to be compensated, Trv_Delay, shall be measured in each transmitted frame at the edge from the EDL bit to the following reserved bit r0, between the edge of the transmitted bit and the edge of the received bit.

The position of the secondary sample point shall be Trv_Delay plus an offset (e.g. half of the bit time in the fast phase), rounded down to the next integer number of time quanta. The resulting secondary sample point may be placed after the end of the transmitted bit. If a bit error is detected at the secondary sample point, the transmitter will react to this bit error at the next following SAMPLE POINT.

6   CAN FD IMPLEMENTATION

CAN FD protocol implementations shall provide the same controller-host interfaces as CAN protocol implementations, to provide an easy migration path for existing CAN applications. The minimum required differences are new configuration registers for the CAN FD operation.

The CAN FD protocol allows frames with more than eight data bytes. It is not required that all CAN FD implementations support longer frames, CAN FD implementations may be limited to a subset of DATA FIELD length. A CAN FD implementation that supports only up to e.g. eight data bytes in a frame shall not treat longer received frames as an error, fault-free longer frames shall be acknowledged and shall take part in acceptance filtering. Received data bytes that exceed the CAN FD's data handling capacity shall be discarded. A such limited CAN FD implementation that is requested to transmit a longer frame shall fill up the data bytes in the frame that exceed the data handling capacity with a constant byte pattern. This pattern shall be chosen so that it does not cause the insertion of STUFF BITS, e.g. 0xCC.

The following optional interface registers provide an extended analysis of the ongoing communication:

* Double set of status registers to distinguish between messages and errors occurring while operating in the first or in the second bit-rate.

* Dedicated error counter to compare error rates in the two operating modes.

* Counters for successful frames to measure the bus traffic.

* Per message status flag indicating whether a message was received using the first or the second bit rate.

* Per message configuration flag controlling whether a message is to be transmitted using the first or the second bit rate.

* Communication management state machine that enables or disables the use of the second bit-rate according criteria like e.g.: Relative error rates in the two bit-rates, reception of message in specific bit-rate, control message received from external bus master, command written by local host.

* Automatic retransmission of failed frames may be limited to a pre-configured value.

* A transmitting node may be required to suspend following transmissions for a pre-configured time after each successful transmission.