CAN FD and the CRC issue

During ISO standardization the original CAN FD protocol needed to be modified, in order to maintain the high level of reliability of Classical CAN. This article illustrates the root cause for this issue as discussed during ISO standardization, shows examples, and describes the solution.

During the standardization processes of CAN FD, a CRC issue was discovered in the ISO/CD 11898-1 version from 2014-08-12. The ISO Task Force decided to solve this issue by changing the FD frame format. This solution has already been included into ISO/CD 11898-1 version from 2014-10-15.

Classical CAN has a known Cyclic Redundancy Check (CRC) weakness. More specifically, in Classical CAN a pair of bit errors in a frame generating/eliminating stuff conditions may reduce the Hamming Distance (HD) to 2. This means that a receiving node may accept a frame and give a positive acknowledge even if this frame has two bit flips. This case was first reported in a paper by Bosch for the ISO task force on CAN (1989) and first published in the SAE paper 900699 by Unruh et al. It is also explained in 1999, "Multi-Bit Error Vulnerabilities in the Controller Area Network Protocol", a thesis paper by Eushian Tran at the Carnegie Mellon University.

In order to prevent CAN FD from inheriting the described CRC weakness, the CRC calculation in CAN FD includes the dynamic stuff bits. However, this change made the CRC calculation more vulnerable to the fault type "shortening or lengthening of the bit sequence" under specific conditions. Cases can be constructed where the receiving node accepts a frame that was falsified by just a single error. This vulnerability in the ISO/CD11898-1 version from 2014-08-12 was detected and solved. The version from 2014-10-15 contains the solution and therefore has a slightly modified FD frame format, which is also robust against shortening or lengthening of the bit sequence. To clearly outline which fault types might occur, we take a look at a fault model. The two considered fault types are fault type A and fault type B.

Fault type A: Bit flip

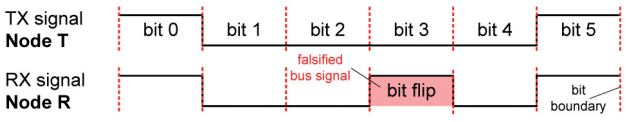


Figure 1: A receiving node samples a bit with the inverse value

Fault type B: Shortening or lengthening of the bit sequence

A receiving node synchronizes on a glitch. Due to this synchronization, in a bit sequence the receiving node may sample one bit less or one bit more than was actually transmitted by the transmitting node. From the receiving node's point of view this is a shortening (one bit less) or lengthening (one bit more) of the bit sequence. Fault type B can also occur in connection with stuff bits. The shortening and/or lengthening may happen several times per frame. Inside one frame either shortening or lengthening is likely to happen. Which one is more likely is determined by the relation between the transmitter's and the receiver's clock rates.

Figure 2 shows an example where the receiving node resynchronizes into the wrong direction due to a falsified bus signal. As a consequence the receiving node samples the bit sequence "100000i" as "100001".

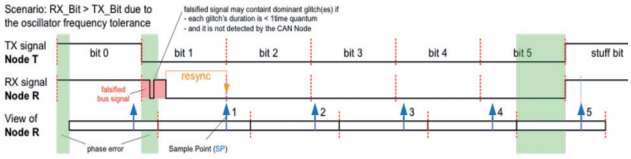


Figure 2: Example for a shortening of the bit sequence

Figure 3 shows an example where the receiving node resynchronizes into the wrong direction due to a falsified bus signal. As a consequence the receiving node samples the bit sequence "100001" as "100000i".

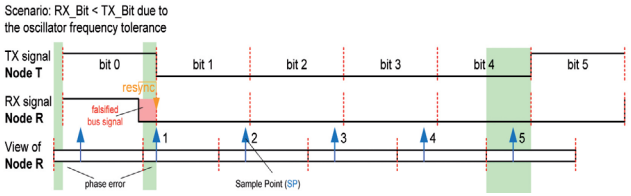


Figure 3: Example for a lengthening of the bit sequence

Root cause for the CRC issue

The receiving node compares the calculated and received CRC bit sequence to decide if a frame can be accepted or not, i.e. if it was received correctly or not. The CRC result is reliable if the CRC-algorithm is applied exactly to the same number of bits (frame length) on sender and receiver side. In this case, the term Hamming Distance can be used. A Hamming Distance of n expresses that frames with up to n-1 falsified bits are detected by the CRC-algorithm as erroneous. The CRC algorithm may or may not detect frames with n or more falsified bits as erroneous.

If the receiving node applies the CRC-algorithm to a different number of bits (less or more) than the transmitting node, the result of the CRC-algorithm has to be regarded as corrupted. This means that a corrupted frame could lead to a positive CRC result too.

Your Source For *SYSTEM INTEGRATION*

Murphy's PowerView displays integrate your control systems and machine data in a powerful, rugged package. With superb environmental specs, bonded screen and sealed connectors, it provides durable performance even in the harshest applications.

PowerView™
Freely Configurable Displays

PV 380

Colour and Monochrome Displays

Freely configurable HMI graphics, state machines, diagnostics, CAN messaging and scripting/activity programming

Compatible with CAN 2.0B, SAE J1939, CANopen, freeform

PV 450

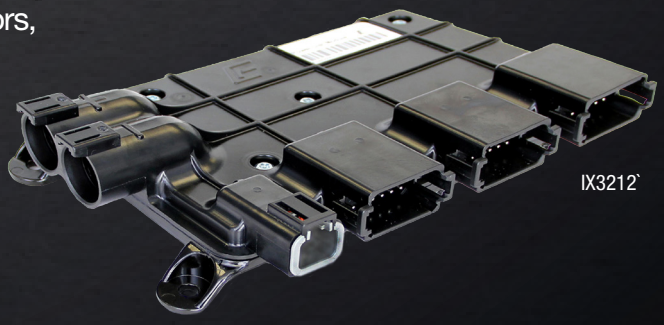
Sunlight Visibility, Works With Polarized Glasses
- Ultra-bright LCD, Bonded to Eliminate Fogging

Robust and reliable: suitable for use outside the cab
- Class-leading environmental protection: IP67, -40°C to +85°C

PV 780

PowerCore™
Controller, Input/Output & Power Distribution Modules

Seamlessly add CAN-controlled solid-state I/O and power distribution where you need it with the PowerCore IX3212. This Power Distribution Module directly drives high-power motors, lamps, actuators and loads via a four-wire power and CAN connection. Improve your control, reliability and load diagnostics while saving on wiring and labour costs.


IX3212`

Reduce wire harness cost with IX3212 Power Distribution Modules:
- 12 outputs: 15A per output/70A per module, on-off/PWM/H-bridge
- 12 digital & 8 analogue inputs

Integrate your total package with Murphy products!

Learn more at *www.fwmurphy.com/ciaintegration*

MURPHY®
by ENOVATION CONTROLS

Or Call: **USA** +1 918 317 2605
UK +44 1722 410055
China +86 21 6237 5885
Latin America +1 918 317 2500
South Korea +82 70 7951 4100
India +91 91581 37633

The consequences for the fault types are:

- Fault type A: The frame CRC is a valid method to detect bit flips. The provided Hamming Distance depends only on the used CRC-polynomial and the length of the frame.
- Fault type B: Due to the shortening and lengthening of the frame, the CRC is corrupted and not sufficient to decide if the frame was received correctly or not. To detect this fault type, the receiver additionally has to know the total frame length, including the number of stuff bits.

As a result, in case of Fault type B the frame CRC is not sufficient to decide if a frame was received correctly or not.

Effect of fault type B on Classical CAN and CAN FD

Classical CAN is able to detect a single error of fault type B with a probability of 100%. To show this, two different cases have to be considered.

Case 1: In Classical CAN, if fault type B occurs at a stuff condition, this does not change the number of bits fed to the CRC algorithm in the receiving node. The receiving node sees a bit flip because in Classical CAN the dynamic stuff bits are not included in the CRC calculation. The frame CRC is sufficient to detect bit flips.

Example for shortening:
TX Bits 0 0 0 0 0 i
RX Bits 0 0 0 0 1 ➔ from the CRC point of view: a bit flip at the 5th bit
Example for lengthening:
TX Bits 0 0 0 0 1
RX Bits 0 0 0 0 0 i ➔ from the CRC point of view: a bit flip at the 5th bit

Legend
blue = bit used for CRC calculation
black = bit not used for CRC calculation
= bit sampled more/less by the receiving node
i = recessive stuff bit

Case 2: If fault type B does not occur at a stuff condition, the received frame length changes and the CRC calculation is corrupted. The receiving node's view is shifted. In case of a single error, the erroneous frame is detected by the frame format checking, because at the end of the frame the receiving node sends its acknowledge either one bit too early or one bit too late.

CAN FD version from 2014-08-12

CAN FD detects a single error of fault type B with a probability of less than 100 %. To show this, two different cases have to be considered.

Case 1: In CAN FD, if fault type B occurs at a stuff condition, this does change the number of bits fed to the CRC algorithm in the receiving node by adding or removing a stuff bit. The CRC calculation is corrupted. Consequently, the frame CRC is not sufficient to decide if a frame was received correctly or not.

Case 2: If fault type B does not occur at a stuff condition, this does change the number of bits fed to the CRC algorithm in the receiving node by adding or removing a bit. The CRC calculation is corrupted. Additionally, the receiving node's view is shifted by one or more bits. If the ratio of data phase to arbitration phase bit-rate is large (e.g. 4 or larger), it is possible that the frame format checking at the end of the frame (CRC delimiter and later) does not detect this shift, because the shift is very small compared to the duration of an arbitration phase bit time.

Which cases need to be covered by the solution?

The residual error rate (Pres) of a communication system is an important attribute. Highly reliable systems require a very low Pres. Pres is a function of the undetected error rate (Pun) of a communication system and the necessary number of errors in a frame to achieve Pun. The more errors in a frame are necessary to achieve a Pun > 0, the lower is Pres.

The solution needed to improve CAN FD so that it has a lower Pres than Classical CAN. To achieve this CAN FD has to fulfill the following requirements:

- CAN FD has to detect frames with a single bit error with 100 % probability, i.e. Pun_CAN_FD = Pun_Classical_CAN = 0;
- CAN FD has to detect frames with two bit errors with higher probability than Classical CAN, i.e. Pun_CAN_FD < Pun Classical CAN. This requirement was already fulfilled in the ISO/CD 11898-1 from 2014-08-12, as the critical bit sequence in CAN FD is much longer than in Classical CAN. The critical bit sequence consists of the bits in the CRC field. The receiving node uses these bits to check if it received the frame correctly.

Accordingly, the solution for the CRC issue needs to detect all single errors in a frame.

Frame bit types

The bits in Classical CAN and CAN FD frames can be divided into two types.

Type 1: Key bits inform the receiver about how to interpret the remainder of the frame. The key bits in CAN FD are: IDE, FDF, res, BRS, and DLC. When one of these bits is sampled inverted in the receiving node, the receiver interprets the remainder of the frame differently than intended by the transmitting node. A frame with two or more falsified bits, where the first is a key bit, might be accepted by the receiving node as a valid frame. Such cases can be constructed in Classical CAN as well as in CAN FD.

Type 2: Non-key bits are all bits except the key bits. If the receiving node samples a non-key bit with inverted value, this has no impact on the interpretation of the remainder of the frame.

Single errors that need to be considered

The solution needs to be able to detect all single errors in a frame. The following list is used to evaluate which single error cases need to be covered by the solution.
Fault type A – Bit flip:

- Key bit: The receiver gets misaligned, consequently the error is detected by the frame format checking. Probability = 100 %;
- Non-Key bit: The bit flip is detected by the frame CRC. Probability = 100 %.

Fault type B – Lengthening or shortening by 1 bit:

- One key bit is sampled inverted: The receiver gets misaligned, consequently the error is detected by the frame format checking. Probability = 100 %;
- All key bits are sampled correctly: The frame CRC is not sufficient to validate the frame. Probability < 100 %, which makes this a CAN FD CRC issue.

The CAN FD CRC issue may occur with fault type B only when all key bits are sampled correctly. The solution for the CRC issue needs to cover those cases where the frame is falsified by a single error that leads to fault type B, when in such an erroneous frame all key bits are still sampled correctly.

Occurrence of the CRC issue

The CRC issue occurs when a CAN FD node (according to the ISO/CD 11898-1 version from 2014-08-12) does not detect a single error in a frame. Consequently it accepts the frame as valid. Both CRC polynomials (CRC-17 and CRC-21) used in CAN FD are affected. The issue can occur at the start of a frame and inside a frame.

At the start of a frame

Cases with falsified Start-of-Frame (SOF) bits were first reported by engineers from Renesas. For these cases it is important to know that the ISO/CD 11898-1 version from 2014-08-12 defines the initialization value for the CRC registers as “0…0”.

Case 1a - Shortening of frame: A local error shortens the SOF bit at the receiving node. Now a frame starting with the identifiers “0000i” may be falsified to “0001”. Figure 4 visualizes the case.

In this case, the bus signal falsification has to be a local

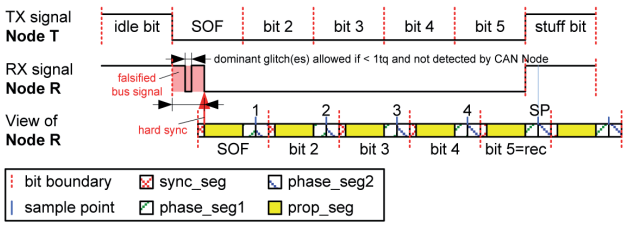


Figure 4: The five dominant bits (SOF + four identifier bits) are sampled as four dominant bits; the subsequent stuff bit is interpreted as a regular bit and not as stuff bit

error. This case can occur independently of the CAN clock frequency relation between transmitter and receiver, because the receiving node performs a hard synchronization at the beginning of the SOF bit. The falsified bus signal may contain dominant glitches, as long as each glitch is shorter than one time quantum and the glitches are not detected by the receiving CAN node.

The required shortening of the SOF bit depends on the CAN clock frequency relation between transmitter and receiver. If the nodes do not have any CAN clock tolerance (df = 0 %) then a shortening/falsification of the SOF bit by “phase_seg2 + ε” is sufficient to create the case. At 1 Mbit/s and a sample point of 80 %, a shortening by 205 ns is sufficient.

Case 1b - Lengthening of frame: A local error lengthens the SOF bit at the receiving node. The lengthening occurs

towards the bit before the SOF bit. Now a frame starting with the identifiers “0001” may be falsified to “0000i”. Figure 5 visualizes the case.

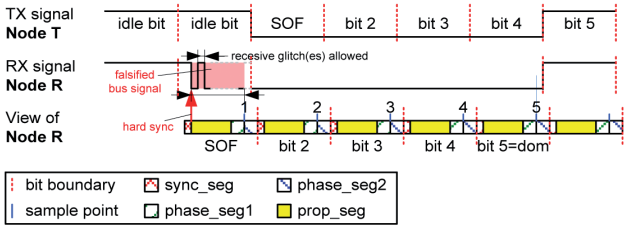


Figure 5: The four dominant bits (SOF + three identifier bits) are sampled as five dominant bits; the subsequent recessive bit is sampled as a stuff bit

In this case, the bus signal falsification has to be a local error. This case can occur independently of the CAN clock frequency relation between transmitter and receiver, because the receiving node performs a hard synchronization at the beginning of the SOF bit. The falsified bus signal may contain recessive glitches, as long as the bit prior to the SOF bit transmitted by the transmitter is sampled as dominant by the receiving node.

Inside a frame

A falsified “0” bit sequence is detected wrongly if it starts when the intermediate CRC register value equals “0...0”. The case can occur at any bit position in between SOF bit and the transmitted CRC checksum. A simple way to construct cases is to exploit the following two facts: Firstly, the intermediate CRC register value may be “0...0” during the reception of a frame. Secondly, in case the intermediate CRC register value is “0…0”, then an incoming 0 (dominant bit) does not change the intermediate CRC register value, while a 1 (recessive bit) does change it.

Case 2a - Shortening of frame: The receiving node samples the bit sequence “00000i” as “00001”. Figure 6 visualizes the case.

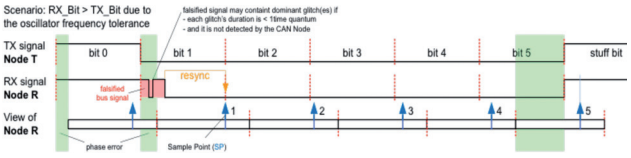


Figure 6: The receiving node samples the bit sequence “00000i” as “00001”

The transmitting node sends the stuffed bit sequence “00000i”. An error shortens the first “0” of this bit sequence. The receiving node resynchronizes on this “wrong” edge.

The intermediate CRC register value has to be “0...0” at the first bit of this sequence. The receiving nodes samples “00001”. This means it samples four “0” bits instead of five. The falsification of the bus signal leads to no correction of the phase error, or even a correction into the wrong direction.

This case may occur, if BitTimeRX_node > BitTimeTX_node due to clock tolerance and if the sample point position in the receiving node is late (e.g. 80 %). The bus signal falsification can be a local error if the transmitting node uses no transmitter delay compensation, or a global error if the transmitting node uses transmitter delay compensation.

This case was reproduced in simulation. Figure 7 shows a simulation screenshot of this case. Focus on the signals “m_can_rx” and “sample point”.

Case 2b - Lengthening of frame: The receiving node samples the bit sequence “00001” as “00000i”. Figure 8 visualizes the case.

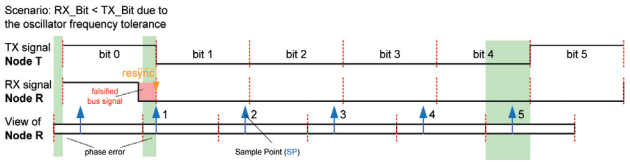


Figure 8: The receiving node samples the bit sequence “00001” as “00000i”

The transmitting node sends the unstuffed bit sequence “00001”. An error lengthens the first “0” of this bit sequence. The receiving node resynchronizes on this “wrong” edge. The intermediate CRC register value has to be “0...0” at the first bit of this sequence. The receiving nodes samples “00000i”. This means it samples five “0” bits instead of four. The “1” in this bit sequence is interpreted as a stuff bit. The falsification of the bus signal leads to no correction of the phase error, or even a correction into the wrong direction.

This case may occur, if BitTimeRX_node < BitTimeTX_node due to clock tolerance and the sample point position in the receiving node is early (e.g. 20 %). The bus signal falsification can be a local error if the transmitting node uses no transmitter delay compensation, or

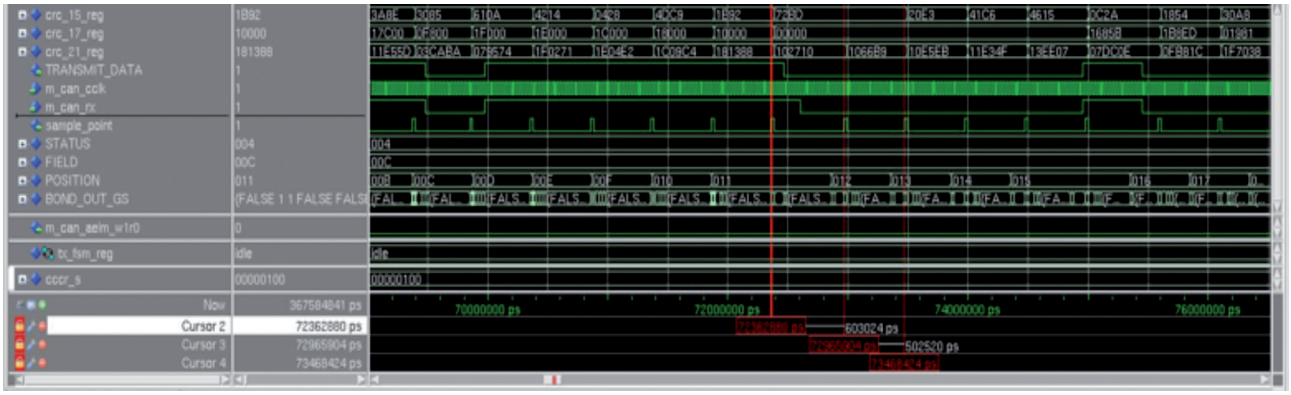


Figure 7: The signals “m_can_rx” and “sample point” are especially interesting

a global error if the transmitting node uses transmitter delay compensation.

Solving the CRC issue

To solve the CRC issue, the receiving node needs to know the total number of transmitted bits for each frame. The receiving node already knows the total frame length excluding dynamic stuff bits from the received frame type and the DLC. Therefore the number of dynamic stuff bits (generated by the Classical CAN stuffing method) is included into the frame format.

It is sufficient to transmit the stuff bit count modulo 8. With this it is possible to detect up to seven lengthening or shortening errors, if these coincide with stuff conditions. This is adequate as we consider a Hamming distance of 6 for CAN FD.

The stuff bit count is transmitted in the first bits of the CRC field, lengthening this field. This position was chosen for two reasons:

Firstly, the stuff bit count needs to be transmitted after switching from dynamic bit stuffing to fixed bit stuffing. In the CRC field the fixed stuff bit method has to be used. At the beginning of the CRC field the nodes know the total amount of dynamic stuff bits in the frame.

Secondly, the stuff bit count needs to be transmitted before the CRC sequence in order to be safeguarded by the frame CRC.

When a stuff bit is dropped or inserted by synchronization failure, the CRC is corrupted. This means the receiver may not be able to detect this error with the help of the frame CRC. When a bit flip falsifies the stuff bit count in the same frame, the receiver may not be able to detect this error.

This makes it necessary to safeguard the transmitted length information itself, i.e. the stuff bit count modulo 8. This is done through the following two safeguards:

- Adding a parity-bit (even parity);
- Gray-coding the stuff bit count.

Table 1: New bits added to the FD frame format and coding of the eight possible stuff bit counts modulo 8

Stuff bit count modulo 8	Bits added to FD frame format		
	Gray-coded value	Parity bit	Fixed stuff bit
0	000	0	1
1	001	1	0
2	011	0	1
3	010	1	0
4	110	0	1
5	111	1	0
6	101	0	1
7	100	1	0

Since the stuff bit count is inserted at the beginning of the CRC field, there is a fixed stuff bit following the parity bit, with the inverse value of the parity bit. Parity check and fixed bit stuffing rules detect any single bit flip of these bits. The same applies for two bit flips if at least one of the bit flips is the parity bit or the following fixed stuff bit. If two bits of the ▷

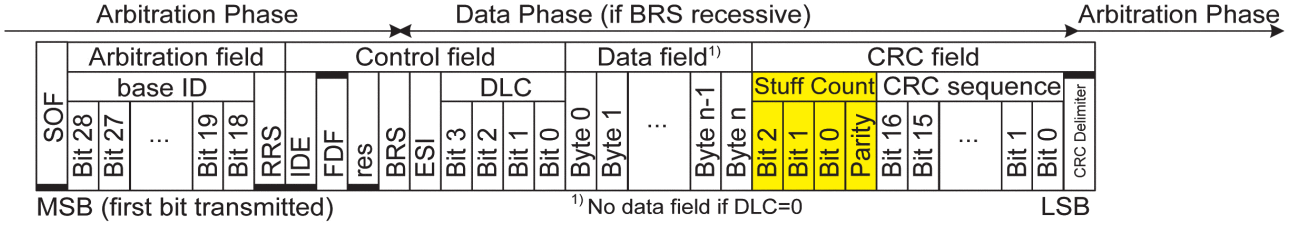


Figure 9: Order of bit transmission in FD base frame format; up to 16 data bytes

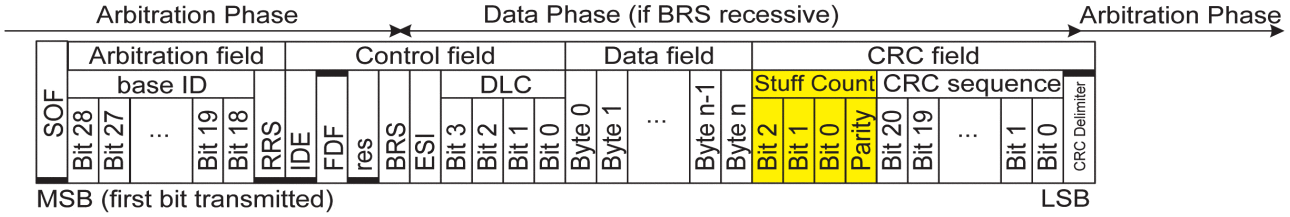


Figure 10: Order of bit transmission in FD base frame format; 20 to 64 data bytes

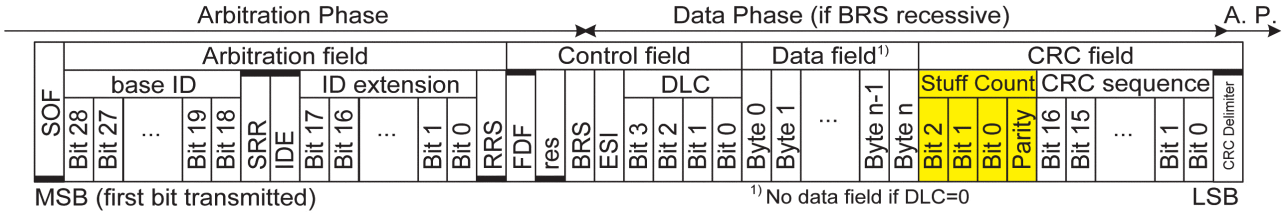


Figure 11: Order of bit transmission in FD extended frame format; up to 16 data bytes

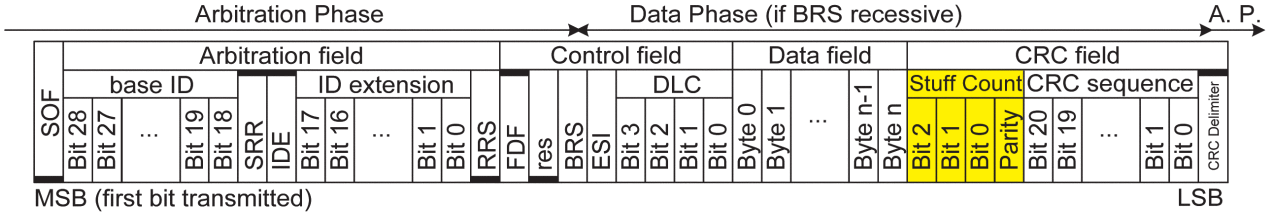


Figure 12: Order of bit transmission in FD extended frame format; 20 to 64 data bytes

gray-coded stuff bit count are flipped, the results is a stuff bit count value with a difference of at least 2, which is detected by a comparison with the internally counted stuff bit value. The minimum number of bit errors that could remain undetected is four, when two bit flips in the gray-coded stuff bit count coincide with two dropped or inserted stuff bits.

CRC17, Stuff count = 0 **S00001**CCCC**S**CCCC**S**CCCC**S**CCCC**S**C
CRC21, Stuff count = 0 **S00001**CCCC**S**CCCC**S**CCCC**S**CCCC**S**CCCC**S**C
CRC17, Stuff count = 3 **S01010**CCCC**S**CCCC**S**CCCC**S**CCCC**S**C
CRC21, Stuff count = 3 **S01010**CCCC**S**CCCC**S**CCCC**S**CCCC**S**CCCC**S**C

Legend: red: fixed stuff bit,
C: CRC bit,
shaded: the 5 new bits inserted to the CAN FD frame

Figure 13: Exemplary bit sequences

The receiver checks the received stuff bit count modulo 8 by comparing the received value with the internally counted value and the parity bit by comparing the received value to the internally calculated value. A mismatch during the comparison is treated the same way as a detected CRC error.

The ISO Task Force also decided to use “10...0” as an initialization vector for CRC-17 and CRC-21. The single “1” is at the most significant bit position. The old initialization vectors were all “0..0”. With this new initialization vector an intermediate CRC register value of “0...0” cannot occur for the first 17 transmitted bits with CRC-17 and for the first 21 transmitted bits with CRC-21.

Changes to the CAN FD frame formats

Figure 9, 10, 11, and 12 show the resulting FD frame formats. The figures do not show stuff bits; changes are highlighted yellow. The part of the CRC field that contains the stuff bit count modulo 8 and the parity bit is named Stuff Count.

Figure 13 shows exemplary bit sequences of the CRC field. Recall that the CRC field starts with a fixed stuff bit. Then a fixed stuff bit is inserted after each 4 bits of the CRC field. The Stuff Count has 4 bit. In total, the CAN FD frame is lengthened by 5 bits, since after 4 bits a fixed stuff bit is inserted into the CRC field.

Author

Dr. Arthur Mutter
Robert Bosch GmbH
www.bosch.de

ifm electronic

Your vehicle can see more than you think...

Visit us at
Hannover Messe 2015
▶ hall 9 · stand D36

...now you can see it too – at a glance.

PDM360 NG-12 dialogue module for mobile machines. Optimum readability thanks to bright 12.1" TFT display with 1024 x 768 pixels and "optical bonding". Backlit function keys with tactile feedback.
Powerful 32-bit controller, freely programmable to IEC 61131-3 with CODESYS.
Internal mass memory and interfaces for USB, Ethernet, CAN and analogue cameras.

www.ifm.com/gb/12inch

Phone: +49 0800 16 16 16 4