**CAN XL error detection capabilities**

With its higher data-rates and payload sizes, CAN XL is the next step in the evolution of CAN. Besides this, CAN XL also provides improved error detection capabilities.

- CAN XL frame format (Source: Bosch)

The complete article is published in the June issue of the CAN Newsletter magazine 2020. This is just an excerpt.

CAN XL offers data-rates and payload sizes that are many times higher than in Classical CAN and CAN FD [1], [2]. Error detection is a crucial functionality provided by communication protocols. A receiving node has to be able to judge if a frame was received with or without errors. Autonomous driving and other safety relevant applications require that frame errors are detected with a very high probability. The acceptance of an erroneous frame should be practically impossible. This article first introduces the three CAN error types known in literature that might occur in a frame in harsh environments: (1) bit error, (2) bit drop and bit insertion, (3) burst errors. The two main pillars of the CAN error detection mechanism are: (A) the cyclic redundancy code (CRC) check and (B) the format checks. Both pillars are strengthened during the currently ongoing specification of CAN XL, to fit to tomorrow’s applications.

We explain how these pillars were improved. Therefor we show the reasons for the chosen CRC concept of having a header CRC and a frame CRC in a CAN XL frame. Further, we introduce the available format checks in CAN XL. Finally, we show systematically how the CAN XL error detection mechanisms master to detect the three error types. A deep dive into the properties and strengths of the used CRC polynomials is given in [9].

**Introduction**

CAN XL is currently being specified inside the CiA’s (CAN in Automation) CAN XL Special Interest Group. The first specification meeting took place in Nuremberg (Germany) on December 17th 2018. The CiA 610-1 specification document, which focuses on OSI layer 2 (known as CAN XL protocol), was not yet finished at the time of writing this article. Consequently, the final CiA 601-1 specification may show differences compared to the content presented in here. [3] gives an overview about the current CAN XL status. Some of the main features of CAN XL are:

- data field size up to 2 048 byte
- gross bit-rate of 10 Mbit/s and more
- strong error detection capabilities

With this set of features CAN enables the usage of higher layer protocols like IP (Internet Protocol). At the same time, it eases the implementation of safety critical applications with its excellent error detection capabilities and its well-known robustness. Two very essential functions in a communication protocol are the error detection and the error handling. They have a large impact on the reliability of the communication system. The focus of this article is the error detection mechanisms in CAN XL.

This article consists of three parts. Part 1 introduces the CAN XL error detection mechanisms and explains how these were improved compared to CAN FD. In this part, the reasons are given for the chosen CRC concept of having a header CRC and a frame CRC in a CAN XL frame. Part 2 introduces the error types known in literature, along with their properties. Part 3 performs a systematic evaluation to show how the CAN XL error detection mechanisms master to detect all known error types up to a given extent.

**CAN XL error detection mechanisms**

In CAN communication, all nodes in a network check the validity of each frame, including the transmitter of the current frame. The checks are based on a combination of several protocol mechanisms for error detection. They are described in the following. Figure 1 shows the current version of the CAN XL frame format. The bits used to implement additional or updated error detection mechanisms (compared to CAN FD) are shaded.

**Bit monitoring**

Bit monitoring means that a node that transmits a bit also monitors the bit values on the CAN network. If the transmitted and received (monitored) bit values differ, the reaction of the node depends on the bit position in the frame. As example, if the transmitting node transmitted a 1 and received a 0 in the data field, it regards this as a bit error. However, if the same happens in the arbitration field, it regards this as arbitration lost.

A detailed explanation of the bit monitoring in CAN FD can be found in [10]. If error signaling (via error frames) is enabled in CAN XL, bit monitoring is nearly equal to that in CAN FD. For the case that error signaling is disabled, bit monitoring is not yet fully specified in the current CiA 610-1 draft.

**Frame format check**

Most parts of a CAN frame (identifier, control, or data bits) are variable or are calculated from the variable bits (CRC sequence), but some bits (delimiters, end of frame) have a fixed format (see figure 1). The bit values of these bits are marked in the figure with a bold line. A receiver detects a form error when it samples a fixed format bit with the wrong value.

A special case is the reserved bit following the XLF bit in CAN XL frames. The reserved bit is expected to be dominant. In current applications, a form error is detected when this bit is sampled as recessive. For future applications, this bit may be used to distinguish between the CAN XL frame format and another – not yet defined – new frame format. When this alternative is selected (by software configuration) and if then this bit is sampled as recessive, the receiver enters a protocol exception state until the network is idle again. This allows the introduction of future new frame formats that are tolerated by existing CAN XL implementations. A node transmitting a CAN XL frame sends the PDF and XLF bits as recessive (logical ‘1’). These bits are part of the arbitration field, which is different compared to CAN FD. This means, if the transmitting node samples one of these bits as dominant, it loses arbitration and becomes a receiver.

In CAN XL, beside the bit-rate, also the mode of the transceiver can be switched. In the error free case, the CAN XL protocol controller signals the mode switch to the transceiver during the bits AL1 and AH1. The signaling of the mode switch to the transceiver, as well as the mode switch of the transceiver may have side effects on the RXD input signal of the protocol controller. Due to this, a CAN XL node does not perform a format check at the fixed format bits (bold lines mark bit value) AL1 and AH1.

**Format check pattern (FCP)**

The FCP field contains only fixed format bits and is used by a receiver for two purposes. The first purpose is that it provides a synchronization edge before the receiver switches from the data phase to the arbitration phase.

The second purpose is that a receiver can check with help of the FPC field if its frame decoding is aligned with the actual transmitted bit position. Disturbed synchronization edges may lead to so called bit insertions and bit drops in the receiver. A receiver can detect, with help of the FPC field, a misalignment of 3 bit in both directions.

- Bit error example (Source: Bosch)

**CRC concept**

In general, the transmitter and the receivers of a frame calculate the CRC (cyclic redundancy check) sequence. After reception of the CRC sequence, each receiver performs a CRC check, to judge if it received the frame correctly or not.
For the CRC's error detection capability to succeed with a very high probability the following two requirements have to be fulfilled:

- RQ1: Transmitter and receiver of the frame calculate the CRC sequence based on the equal number bits.
- RQ2: The receiver checks the CRC sequence at the right position inside the transmitted frame.

To fulfill RQ1, the CAN XL frame format uses fixed stuff bits in nearly the whole frame. Dynamic stuff bits are only used in the first bits of the header, to be compatible to CAN FD. A bit insertion or drop error at a dynamic stuff condition changes the number of bits fed into the CRC. As the error just adds or removes a dynamic stuff bit, the format checks described up to now cannot detect that error. With fixed stuff bits, the frame has a defined length in bits and the receiver can feed the exact number of bits into the CRC calculation.

To fulfill RQ2, we need to make sure that a transmission error cannot change easily the position, where the receiver expects the CRC. For example, if the DLC (data length code) is falsified, the receiver checks the CRC at a wrong position. To solve this, CAN XL uses, like Flexray, a header CRC, and a frame CRC. The header CRC safeguards a header of well-known length. If a receiver saw a valid header CRC, it is very likely that the DLC is correct. With the correct DLC, the data field length is also well known.

Scope of the frame CRC

The frame CRC is calculated over the header and the data field (see figure 1), which is similarly done in Flexray. The author in [9] describes in detail, which bits are included and which are excluded from CRC calculation. This “double checking” of the header is done, because on the one side the frame CRC performance is practically not weakened by safeguarding these few additional header bits. On the other side, “double checking” increases the probability to detect transmission errors in the header, which were not detected by the header CRC.

If you would like to read the complete article you can download it free-of-charge or you download the entire magazine.