Migrating from Flexray to CAN XL

The VW Group has started to consolidate its EE architectures towards E²2.0 for all types of cars, e.g. volume, premium, sports, etc. A solution is required to have one communication technology for all vehicles, together with some other handy features. This article describes the challenges and gained possibilities of migrating from Flexray to CAN XL networks using the powertrain example, also considering CAN FD and why it is only an intermediate solution.

Flexray [1] was designed to provide a high-performance networking technology to cope with time-critical communication demands in drive and powertrain control loops between ECUs (electronic control units) distributed throughout the car. Due to its complexity, limitations, and involved costs, it only found its use in premium cars. IC (integrated circuit) supplier support as well as OEM (original equipment manufacturer) market acceptance was not overwhelming.

Flexray started to enable time-triggered fast network communication for use cases like x-by-wire. These use cases required redundancy by concept, a synchronized system by design to enable precise causal computation and control loops in drive and the powertrain sub-systems as well as the first steps for assisted driving. In order to be safe and reliable, the system requires a fledged specification, where almost no room for interpretation should be given.

On the other hand, OEMs wanted design flexibility regarding communication schedule, redundancy, topology (e.g. active star couplers), etc. This led to vast amount of configuration parameters and cross dependencies, making it severe complex and expensive to handle. Furthermore, the physical layer was a challenge and had to be limited due to some design constraints, such as the majority voting and configurable receiver thresholds leaving almost no room for flexible networks under generalized configuration set-ups.

Finally, the daisy chaining of networks brought limitations in flexibility in production when it came to flashing and calibrating of ECUs. If not all ECUs were assembled to the vehicle at the same time and simple parameters such as bus terminations could not be met.
The CAN FD option

One option to solve the above-mentioned cost problems and the demand of system design flexibility is the CAN FD [2] approach for faster bit rates and larger payloads than CAN CC (classic) to still provide assisted driving and faster causal computation and control loops in a reasonable way for volume cars. But CAN FD does not feature a 256-byte payload as Flexray provides. Additionally, the limitation to 500 kbit/s in the arbitration phase when using CAN SIC transceivers and the maximum possible data phase bit rate of 5 Mbit/s are not sufficient for future application demands.

Prior to the introduction of CAN FD networks in VW Group cars (volume and premium) in 2019, many investigations have been conducted to check the system behavior for networks running in different configurations between 500 kbit/s and the desired and reasonable bit rate of 5 Mbit/s. It turned out, that networks in (multiple)star configuration for the most flexible use with 5 Mbit/s were extremely sensible and prone to ringing in areas, where a system designer does not want them to be.

It also turned out, that reducing network flexibility down to daisy chains adds other types of issues (e.g. plateaus in the area of receiver thresholds), see Figure 2 showing the TXD and $V_{\text{DIFF}}$ signal of a high-ohmic ECU inside the daisy chain at a desired network topology size about to turn over the 0.5-V threshold.

The total line length and number of nodes had to be stripped down to a very short end, which led to a refusal of CAN FD networks (without SIC transceivers) using 5 Mbit/s in our systems. Thus, judging by the signal quality in those systems, no alternative fallback or future-proof solution.

The CAN XL alternative

Considering the above-mentioned requirements and the CAN FD limitations, it seems that CAN XL is a possible alternative for Flexray. The maximum bit rate of up to 20 Mbit/s when using CAN SIC XL transceivers in FAST mode and the 2048-byte payload of the CAN XL data frames overcome the limitations of CAN FD networks, also those using CAN SIC transceivers.

A practical example shows, how a network system migration towards CAN XL can be done. This takes an already transferred communication system example from Flexray to CAN FD with CAN SIC transceivers. Sharpening the focus, two example CAN FD networks (with SIC transceivers) are taken into account, which could represent the split of a real-time and additional data communication for a system design within a possible drive or power-train sub-system.

The goal is a real-time communication having these two network topologies consolidated back on one bus, which required to be split up from a Flexray system design.

This consolidated bus takes both communication set-ups into account and is furthermore designed to work from a physical layer perspective under all environmental conditions and specification limits.

For this, two different analyses are done. The first covers the consolidation of the communication on protocol layer by calculations and the second covers the feasibility of the physical layer by simulations.
Protocol layer analysis

One of these sibling networks sends real-time data for fast control-loop precision. The other bus takes all the "organizational" and "safety/security" related overhead. Within this context several meanings and system measures are used:

- **Real-time**: A communication loop shall be less than 5 ms, where data is available between a given sender/receiver combination in a repetitive manner;
- **Message cycle**: Repetitive transmission loop of a message;
- **Cycle jitter**: Timing variation of the message cycle based on best and worst case prioritized bus access;
- **Latency**: Time for the transmission of a message from message buffer at a sending node over the network topology to the message buffer of a receiving node;
- **Minimum latency**: Latency value for the transmission of a message with immediate and undisturbed bus access;
- **Worst-case latency**: Latency value for the transmission of a message with worst case bus disturbance;
- **Bus-load**: Average time of active message transmission/communication time over a given time period.

All of the above-described system measures are affected by the configuration of the active communication time on the bus and its detailed configuration, which shall be kept as low as reasonable and equally distributed. Equal distribution at first is affected by the differing size of message payloads and number of messages in conjunction. Furthermore, equal distribution requires a proper scheduling of message cycle loops in a communication set-up for a bus system inside the participating microcontrollers in the ECUs, meaning not all message cycles start at a defined cycle start point in time. How to achieve this, will not be discussed in this article.

The bus-load is affected by the number of messages sent on the bus, as well as by the length of data to be transmitted in each message and the used cycle time for each message. When using CAN CC or CAN FD, the arbitration length via the use of 11-bit or 29-bit IDs is also affecting the bus-load. Using the CAN XL protocol, only priority IDs with a fixed length of 11 bit are used [2]. Less messages for less arbitration overhead, lower message cycles and less message payload data basically lead to lower bus-loads.

The latency is affected by the number of messages sent on the bus and especially the length of data in each message, together with its priority identifier and message cycle. Less messages for less arbitration overhead and losses as well as smaller message payloads basically lead to lower latencies, respectively lower minimum and worst-case latencies. Worst-case latencies together with message cycle times directly affect the cycle jitter.

Message payloads using CAN FD are additionally affected by the limited and not byte-wise configuration of the DLC. Payloads above 8 byte almost always added fill bytes to conform the DLC configuration in case of signal data being not a multiple of 4 (up to 24 byte) or 8 (up to 32 byte) or 16 (up to 64 byte). This limitation is not existing in CAN XL and byte-wise configuration is possible. The derivation of the formulas behind the calculation of the latencies, bus-load, and cycle jitter are based on a VW Group internal paper [3].

Physical layer analysis

For the physical layer analysis, a reference model for mixed signal (analog, digital) simulation has been developed, which can be configured in different manners to walk on the physical layer specification limits, given in [2]. This includes for example output driver slopes and amplitudes, as well as transmitter and receiver asymmetries. Further PMD (physical medium-dependent) sublayer and transmission line components were taken from existing simulation ecosystems for CAN SIC and Flexray networks.

System termination values have to be derived based on the number of expected ECUs and total line length in one network topology to stay within physical layer specification limits. This is not being discussed in this article.

Within the physical layer context, several meanings and system measures are used:

- **Signal integrity**: Describing the necessary level of signal quality at the CAN pins analyzed by appropriate qualification criteria;
- **Eye diagram**: Single bit-wise qualification criterium on the convoluted differential signal between the CAN pins.

The signal integrity is evaluated by the measurement of the eye diagram at each sender/receiver combination in the network topology at each receiver, except the sender itself. The eye diagram is configured by means of the possible asymmetries, passive parasitic effects (based on the expected network topology size in terms of number of ECUs and total line length) and tolerances in the system, safety margins for EMC (electro-magnetic compatibility) robustness and bit-timing configuration.

The maximum eye opening is at the sampling point, which is configured to be in the middle of the bit plus one time-quanta. The calculation of the eye-opening values mentioned above is not part of this article.

Many analyses via simulation testbenches, lab, and EMC chamber measurements have shown, that the switching between SLOW and FAST mode in CAN XL networks turns out to be less as critical as the pure signal integrity at the FAST mode. However, the switching from SLOW to FAST mode has still an impact on the asymmetry on the first FAST mode bit pulse on the bus, a reasonable PCS (physical signaling sublayer) stimulus implementation is necessary for simulation-based analysis.

Communication set-up analysis

Flexray as a time synchronous bus system has regularly a reasonable message cycle of 10 ms or less with theoretically no cycle jitter, meaning the worst-case latency basically being equal to the minimum latency.

By the arbitration scheme used in CAN-based asynchronous bus systems with prioritized bus access, cycle jitter is very likely to happen and worst-case latency...
itself as well as cycle jitter must be limited, even when keeping fast message cycles equal to or less than the real-time window.

The premise for a possible consolidation shall be to keep the quality of the real-time communication besides the additional communication demand in one single CAN XL network from the former sibling networks in terms of the initial bus facts. Furthermore, there is room for future extensions of the target CAN XL bus, if further functionalities or ECUs are added to the network.

The message amount sent to the data bus is around four times the value of the real-time bus. The bus-loads are in the range of approximately 24 % to 29 % with approximately 18 % to 22 % overhead by arbitration.

The worst-case latency at the real-time bus calculates to 2 ms and at the data bus to 10 ms. The minimum used message cycle at the real-time bus is 5 ms and the maximum used message cycle is 200 ms, whereas the numbers for the data bus are 10 ms and 1000 ms. The maximum used payload at the real-time bus is 24 byte or less and at the data bus 64 byte or less.

The cycle jitter at the real-time bus reaches 30 % especially at messages with low message cycle time and low priority identifier due to many fast messages with very low message cycle times. In contrast, the cycle jitter at the data bus is approximately the half of the value due to no messages with very low message cycle time, blocking the transmission of other messages with lower priority.

The cycle jitter at the real-time bus reaches 30 % especially at messages with low message cycle time and low priority identifier due to many fast messages with very low message cycle times. In contrast, the cycle jitter at the data bus is approximately the half of the value due to no messages with very low message cycle time, blocking the transmission of other messages with lower priority.

The first step to one CAN XL bus is the raw merge of the communication set-ups from both initial CAN FD networks. With this raw merge, the arbitration rate will be kept at 500 kbit/s and bit rates from 10 Mbit/s to 20 Mbit/s are analyzed. In Figure 6, the initial comparison of the raw merger from the sum of both CAN FD networks (light grey bars) to CAN XL with 10 Mbit/s (light violet bars) and 20 Mbit/s (dark violet bars) is given.

This first comparison shows, that as expected, the bus-load increases heavily due to the combined number of messages from both initial CAN FD networks. This further implies, that the bus-load overhead by arbitration increases accordingly and the worst-case latencies for the data bus related messages increase by some milliseconds. The worst-case latency for the real-time data remains around 2 ms or slightly better and the cycle jitter keeps the same dimension.

To possibly gain more potential with CAN XL communication, another paradigm in the communication set-up design should be analyzed in contrast to nowadays parameter-based communication with probably unchanged message routings throughout a vehicle and equal cycles for application and message transfer. Two different optimization approaches exemplarily show, how this can be achieved. Both initial busses operate with six ECUs, which plays a role at the discussed optimization steps.

### Optimization

**The first approach** focuses on reducing the bus-load by a consolidation of signal data sent by one ECU into only a few messages. This means, that at first data with nearby message priority and cycle time is merged into as less messages as possible.

As a premise for the consolidation of signal data, only application-based data are merged. Other parameter data, e.g. for network management, remain untouched.

To reach a bus-load as low as possible, a good balance should be found with regard to which messages are consolidated into each other to:

- Not send too much data too fast and
- Not send too much low-priority data with too high priority

This prevents too long messages with too high priority on the bus and in general an equally distributed message size.

Figure 7 (for 10 Mbit/s) and Figure 8 (for 20 Mbit/s) show the comparison between the sum of both CAN FD networks (light grey bars) to CAN XL as raw merger (light violet bars) and as bus-load optimized set-up (dark violet bars).

The results after the bus-load-based optimization show that the achieved bus-load is heavily reduced, compared to the raw merging of the CAN-FD networks and is even below the bus-load of one single CAN FD network.
Consecutively the overall worst-case latency of the data from the initial data bus improved to approximately 2.4 ms to 3 ms (depending on the CAN XL bit rate) and for the data of the initial real-time bus to 1.4 ms to 0.9 ms (depending on the CAN XL bit rate), which means that all transmitted data has real-time latency and very stable cycle jitter of even less than the half of the raw merged bus.

With the consolidation through the first approach, the maximum payload reaches approximately 180 byte, which is almost three times the maximum of CAN FD. The number of messages was reduced to nearly a quarter of the whole initial communication set-up.

The second approach focuses on reducing the worst-case latencies even more. This is achieved by drastically reducing the number of sending application messages to one. This implies, that all application data of each ECU is sent with the fastest message cycle and highest bus priority from the original data set of the initial busses. The results of the second approach slightly differ compared to the first approach. The overall worst-case latencies could be further reduced by approximately 0.5 ms. The worst-case latencies for the real-time data remained the same. Due to the above-mentioned fact, sending all data with highest priorities and fastest message cycles, the maximum payload increased to around 360 byte.

The bus-loads and cycle jitters remained the same at 20 Mbit/s, whereas slightly improved by 2 % to 3 %. This shows, that the “one message for all” approach improves best by increasing the bit rate.

Physical layer signal quality

Additionally, the network topology must work under the desired bit rates and further premises from the physical layer perspective, such as the wire-harness topology design needs to be as flexible as with CAN CC networks avoiding daisy-chain or linear topologies.

The network topologies for drive and powertrain sub-systems are by their nature mainly located in the area around the moving axles of a vehicle, far from the passenger cabin. Scalable electric powertrains exemplarily have accelerating and decelerating parts at the front and the rear axle, such as one or more electric motors and possibly further active differentials or torque distributors and braking system assembled. Finally, they are somehow connected to a central computing instance.

Initial example topologies are built up as 5-Mbit/s daisy-chain networks with six ECUs and a 20-m total line length each and additional six inline connectors due to exemplary modular part assembly.

The network topologies for drive and powertrain sub-systems are by their nature mainly located in the area around the moving axles of a vehicle, far from the passenger cabin. Scalable electric powertrains exemplarily have accelerating and decelerating parts at the front and the rear axle, such as one or more electric motors and possibly further active differentials or torque distributors and braking system assembled. Finally, they are somehow connected to a central computing instance.

Initial example topologies are built up as 5-Mbit/s daisy-chain networks with six ECUs and a 20-m total line length each and additional six inline connectors due to exemplary modular part assembly.

The second approach focuses on reducing the worst-case latencies even more. This is achieved by drastically reducing the number of sending application messages to one. This implies, that all application data of each ECU is sent with the fastest message cycle and highest bus priority from the original data set of the initial busses. The results of the second approach slightly differ compared to the first approach. The overall worst-case latencies could be further reduced by approximately 0.5 ms. The worst-case latencies for the real-time data remained the same. Due to the above-mentioned fact, sending all data with highest priorities and fastest message cycles, the maximum payload increased to around 360 byte.

The bus-loads and cycle jitters remained the same at 20 Mbit/s, whereas slightly improved by 2 % to 3 %. This shows, that the “one message for all” approach improves best by increasing the bit rate.

Physical layer signal quality

Additionally, the network topology must work under the desired bit rates and further premises from the physical layer perspective, such as the wire-harness topology design needs to be as flexible as with CAN CC networks avoiding daisy-chain or linear topologies.

The network topologies for drive and powertrain sub-systems are by their nature mainly located in the area around the moving axles of a vehicle, far from the passenger cabin. Scalable electric powertrains exemplarily have accelerating and decelerating parts at the front and the rear axle, such as one or more electric motors and possibly further active differentials or torque distributors and braking system assembled. Finally, they are somehow connected to a central computing instance.

Initial example topologies are built up as 5-Mbit/s daisy-chain networks with six ECUs and a 20-m total line length each and additional six inline connectors due to exemplary modular part assembly.

The second approach focuses on reducing the worst-case latencies even more. This is achieved by drastically reducing the number of sending application messages to one. This implies, that all application data of each ECU is sent with the fastest message cycle and highest bus priority from the original data set of the initial busses. The results of the second approach slightly differ compared to the first approach. The overall worst-case latencies could be further reduced by approximately 0.5 ms. The worst-case latencies for the real-time data remained the same. Due to the above-mentioned fact, sending all data with highest priorities and fastest message cycles, the maximum payload increased to around 360 byte.

The bus-loads and cycle jitters remained the same at 20 Mbit/s, whereas slightly improved by 2 % to 3 %. This shows, that the “one message for all” approach improves best by increasing the bit rate.

The remaining star-based network topology design for CAN XL is comparable to the CAN SIC network topologies in terms of total line length and in-liners, due to the exposed areas and modular assemblies of the ECUs.

In parallel to the communication set-up analysis, the derived CAN XL network topology is executed in simulation set-ups from 5 Mbit/s to 20 Mbit/s and evaluated by the signal-integrity measures discussed above. The creation of...
of the eye-diagram signals is done directly by a simulation-model implementation to provide a correct synchronization of the given signals with an emulated PCS for worst-case stimulus according to [2] as only a little overhead in the executable testbench without having a complex post processing.

Figure 11, Figure 12, and Figure 13 show the worst-case signal integrity found in the network topology with worst-case transmitter asymmetry and weakest transmitter output driving values and an eye opening considering 200 mV additional EMC safety margin on top of the worst-case receiver thresholds given by [2]. The differential voltage signal is shown as the bold line, the synchronized eye-diagram raw signals are shown as dotted lines.

Figure 11: Worst case signal integrity at CAN XL network topology at 10 Mbit/s (Source: CARIAD SE)

Figure 12: Worst case 1 signal integrity at CAN XL network topology at 20 Mbit/s (Source: CARIAD SE)

Figure 13: Worst case 2 signal integrity at CAN XL network topology at 20 Mbit/s (Source: CARIAD SE)

According to the results it can be said, that at lower bit rates the signal reflections implied by the network topology layout can shrink bit amplitudes in the same bit after initiating a slope, basically stressing the sample point. Whereas at the faster bit rates the signal reflections are jumping over to the following bits. This further implies, that depending on the number of sent bits of the same logical bit level have a huge impact on the signal shapes, see Figures 19 and 20. Simulations with bit rates between 10 Mbit/s and 20 Mbit/s show the overlapping of these effects from the corner case bit rates, but still keeping outside of the eye.

Together with in-vehicle EMC measurements of similar network topologies in terms of number of nodes and total line lengths, it can be said that the given example is capable of running at any bit rate up to 20 Mbit/s.

The physical-layer simulations show, that any of the bit rates up to 20 Mbit/s are visible, even under worst-case conditions and flexible star-based network topology designs.

**Conclusion**

Summing up all of the previous analyses, using CAN XL makes Flexray obsolete, reduces the number of necessary network interfaces in central computing or zonal ECUs, reduces the number of network interfaces in real-time sensing/acting ECUs, reduces the number of wiring harnesses, omits private direct connections between real-time sensing/acting ECUs, and omits oversized automotive Ethernet-based and switched multiple point-to-point connections in real-time vehicle movement and control loops. This leads to an overall reduction of development and system complexity, material use, vehicle weight, and therefore to less costs. All of the above-mentioned benefits lead to a lot of headroom in network design in terms of number of nodes, total line length and bit rates for being a future-proof technology.

The heavy improvement of flashing time through faster bit rates and larger payloads just comes as a bonus factor by the technology itself, as well as built in Ethernet frame mapping, routing, tunneling. Adding functionalities can happen with a stable and powerful physical layer, without adding more and more networks to the network architecture. This helps even more when thinking about development concepts such as software-defined vehicles (SDV).

Based on the analysis described above, the communication set-up analysis in particular, focused on the pure consolidation of parameters with the drawback of probably sending some data more than necessary. A PDU-based communication set-up as a further optimization step must be analyzed. Only transmit data, when needed and not all at once furthermore reduces the calculated bus-load and latencies giving even more headroom for a stable and future-proof network design.

(This article is based on the 18th international CAN Conference (iCC) keynote presentation by Marko Moch. The complete paper is published in the 18th iCC proceedings 2024; CiA, Nuremberg.)

**References**


**Author**

Marko Moch  
CARIAD SE (VW Group)  
marko.moch@CARIAD.technology  
www.CARIAD.technology