

# CAN Newsletter Online

CAN NEWSLETTER MAGAZINE

## Migration to CAN FD

In 2011 efforts were made to overcome the existing limits of the Classical CAN in terms of the maximum achievable bit rate – the idea of CAN FD was born. But is it actually worth migrating to CAN FD?

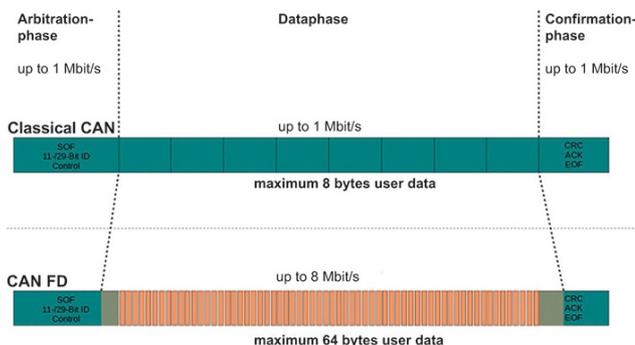
The [complete article](#) is published in the [December issue](#) of the *CAN Newsletter magazine 2021*. This is just an excerpt.

As with the Classical CAN protocol it was the automotive industry that stood as a driving force behind the development of the CAN FD protocol (CAN with flexible data rate). In cooperation with some other experts Bosch began working on a solution in 2011 to shift the existing limits set by Classical CAN regarding the maximum available bit rate respectively the maximum achievable data throughput. At the same time, it was a declared aim to preserve the proven concepts of Classical CAN, such as real-time bus arbitration, event control, 11-bit and 29-bit CAN-Identifier, and multi-manager capability. Moreover, a high robustness against interference, low power consumption and the use of existing topologies were further advantages that needed to be maintained.

The desired objectives were achieved:

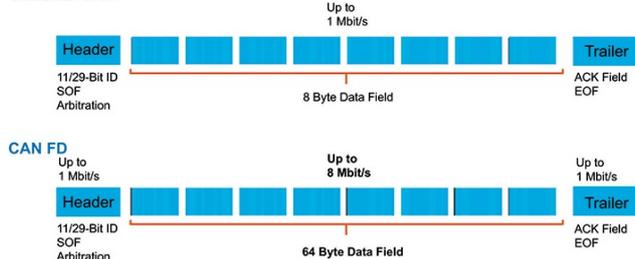
- Maintaining the Classical CAN concepts of arbitration and confirmation phase as well as error management
- Increasing the bit rate during the data phase from a maximum of 1 Mbit/s up to 8 Mbit/s and more
- Increasing the number of transferred data bytes transmitted in a CAN frame from maximum 8 bytes to maximum 64 bytes

The new protocol has been published as an international standard since 2015 with all CAN FD controllers being backwards compatible and still supporting the Classical CAN protocol. A wide range of dedicated CAN FD controllers, micro-controllers with integrated CAN FD interfaces, and FPGA-based (field programmable gate array) solutions are nowadays available.



The comparison of protocol framework (Source: ESD Electronics)

### Classical CAN



The data fields in comparison (Source: ESD Electronics)

### Protocol details

In Classical CAN the transmission of a frame can be divided roughly into three phases: bus arbitration, data transfer, and confirmation. During all these stages, bits are being transferred with an identical bit rate, while all network participants resynchronize constantly in order to compensate for phase noise and phase drift of independent local oscillators. This is especially important during the arbitration and confirmation phases, since all nodes must be broadcasting simultaneously on the network, and each individual node must be able to compare its sent bit with those of other participants. This property of the Classical CAN

protocol determines the physical limits for the maximum possible bit rate or cable length.

The idea behind the CAN FD protocol is to send data with a second, usually much higher bit rate during the data phase. Post-synchronization is suspended during this phase since, due to the principle, there should be only one transmitter on the bus. Furthermore, the payload of a frame has been increased from 8 byte to 64 byte ensuring a considerable improvement in the ratio between protocol and user data. Bit rates with a ratio of 1:4 between arbitration phase and data transfer phase result in an increased net data rate by the factor of 2 up to even 5 depending on the payload size.

To implement the CAN FD protocol a previously reserved bit within the control field of the CAN frame was used, respectively two more bits were added:

- Extended data length (EDL)
- Bit rate switch (BRS)
- Error state indicator (ESI)

A CAN controller will recognize the CAN FD format with the help of the recessive EDL bit (dominant and unused in the Classical CAN protocol). The newly-added BRS bit determines whether the higher bit rate will be used in the data phase or the frame continues to be sent at the arbitration bit rate. Finally, the ESI bit with its dominant status indicates that the sender is in the error active state. On grounds of efficiency the size of the DLC (data length code) field with 4 bits has been left unchanged so that CAN FD frames with more than 8 data bytes can only be sent in discrete quantities. In order to achieve the same degree of robustness against communication errors despite pro-longed payload sizes, in CAN FD a 17-bit checksum (frames with up to 16 bytes of user data) or a 21-bit check-sum (frames with more than 16 bytes of user data) is used to check correctness instead of the usual 15-bit checksum (Classical CAN).

DLC	Classical CAN			CAN FD			Classical CAN			CAN FD		
	User Data	Checksum	DLC	User Data	Checksum	DLC	User Data	Checksum	DLC	User Data	Checksum	
0	0	CRC-15	0	CRC-17	8	8	CRC-15	8	CRC-17	0	0	
1	1	CRC-15	1	CRC-17	9	8	CRC-15	12	CRC-17	1	8	
2	2	CRC-15	2	CRC-17	10	8	CRC-15	16	CRC-17	2	16	
3	3	CRC-15	3	CRC-17	11	8	CRC-15	20	CRC-21	3	24	
4	4	CRC-15	4	CRC-17	12	8	CRC-15	24	CRC-21	4	32	
5	5	CRC-15	5	CRC-17	13	8	CRC-15	32	CRC-21	5	40	
6	6	CRC-15	6	CRC-17	14	8	CRC-15	48	CRC-21	6	48	
7	7	CRC-15	7	CRC-17	15	8	CRC-15	64	CRC-21	7	64	

A comparison of the data length code (DLC) for CAN and CAN FD

Table 1 provides an overview regarding the assignment of the DLC to the amount of data transferred and the checksum used in each case. However, the RTR (remote transmission requests) feature is no longer supported in the CAN FD protocol. Due to the backward compatibility, the use of RTRs is still supported by CAN FD controllers for the Classical CAN protocol.

Both main innovations introduced by the CAN FD protocol – higher bit rate in the data phase and frames with up to 64 data bytes – can be used in a wide variety of ways in applications:

- Improved data throughput is particularly noticeable when transferring large data sets (such as firmware updates)
- Improved real-time behavior (reduction of latency) with higher bit rates during the data phase and unchanged protocol
- Reduced bus load with higher bit rates during the data phase and unchanged protocol allows extensions in systems that otherwise would not be expandable due to the current bus load and might have required an additional CAN network.
- Simple safeguarding of data consistency regarding process data of more than 8 data bytes that can now be sent in a frame with more data bytes
- Possibility to extend existing CAN networks that are already operating at their physical limit regarding the bit rate. This is done by reducing the arbitration bit rate and using a higher bit rate in the data phase, so that an identical or even larger net data rate is achieved.

While the CAN protocol itself is already quite robust the improved protection against transmission errors by means of extended checksums as well as the indication of the sender's state (error passive or error active) are additional advantages that make the communication even more secure.

*If you would like to read the full article, you can [download](#) it free of charge or you [download the entire magazine](#).*

[CW](#)