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?

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.

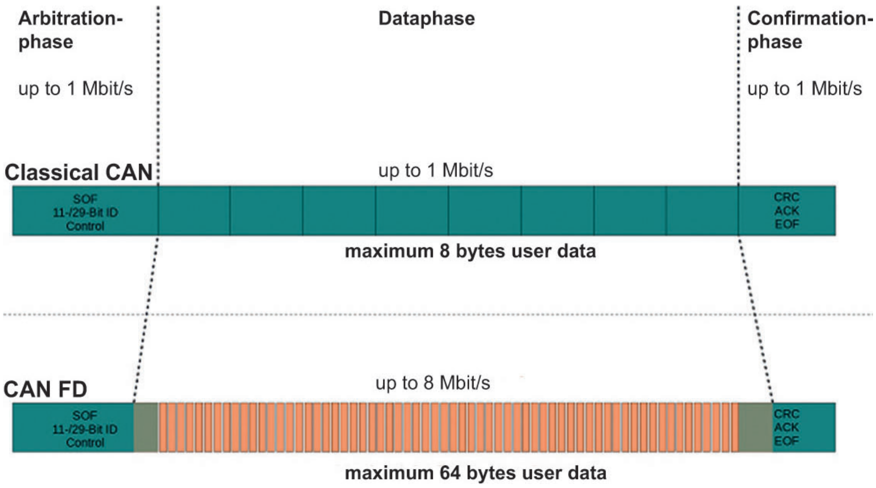


Figure 1: The comparison of protocol framework (Source: ESD Electronics)

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.

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 ▷

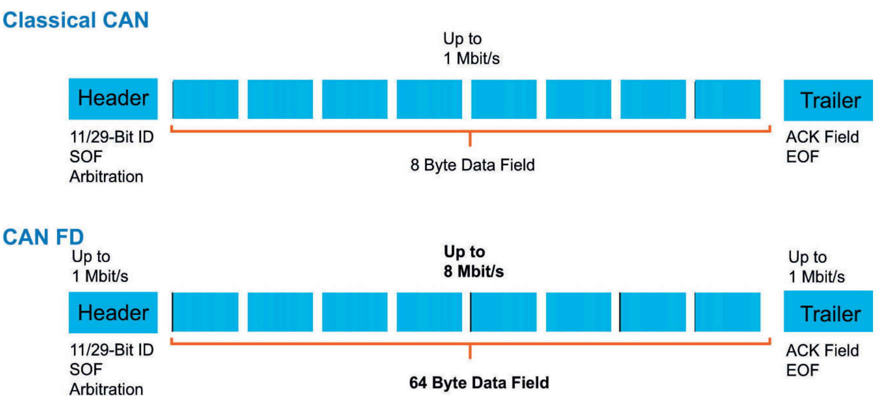


Figure 2: The data fields in comparison (Source: ESD Electronics)

Table 1: A comparison of the data length code (DLC) for CAN and CAN FD

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

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 prolonged payload sizes, in CAN FD a 17-bit checksum (frames with up to 16 bytes of user data) or a 21-bit checksum (frames with more than 16 bytes of user data) is used to check correctness instead of the usual 15-bit checksum (Classical CAN).

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.

Advantages of CAN FD

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.

Performance gain

Any statement regarding the expected gain in throughput or the reduction of latency when switching from Classical CAN to CAN FD is not easy. It basically depends on the type of implementation and the higher-layer protocol used as well as on other boundary conditions. For a better estimate, Table 2 shows the bits required for the transmission of frames with 11-bit CAN-Identifiers at different data lengths and different ratios between the bit rate in the CAN FD arbitration and data phases. The number of data bits refers to bit times of the arbitration phase: i.e. this number indicates the CAN frame length (time) expressed in multiples of 1 arbitration-phase bit time. Based on Table 2, two extremes will be considered when trying to estimate the effort involved for switching from CAN to CAN FD.

Table 2: Performance of CAN and CAN FD for various bit ratios

	Classical CAN	CAN FD			
Ratio bit rates	N/A	1	2	4	8
Data bytes	Data bits*				
1	55	64	47	38	33
2	63	72	51	40	34
3	71	80	55	42	35
4	79	88	59	44	36
5	87	96	63	46	37
6	95	104	67	48	38
7	103	112	71	50	39
8	111	120	75	52	40
12	*198***	152	91	60	44
16	*222***	184	107	68	48
20	*309***	220	125	77	53
24	*333***	252	141	85	57
32	*444***	316	173	101	65
48	*666***	444	237	133	81
64	*888***	572	301	165	97

* 11-Bit CAN Identifier without Stuffbits
** Transmission as subsequent classical CAN messages

Conversion from CAN to CAN FD

Without changing the protocol: Besides the need to purchase CAN FD capable hardware, the (software) effort is in the best case limited to setting the increased bit rate in the data phase. With a ratio of 1:4 between arbitration and data phase bit rate and with a protocol based primarily on frames with 8 data bytes, one can expect to improve the throughput by a factor of more than 2 or to halve corresponding latency times.

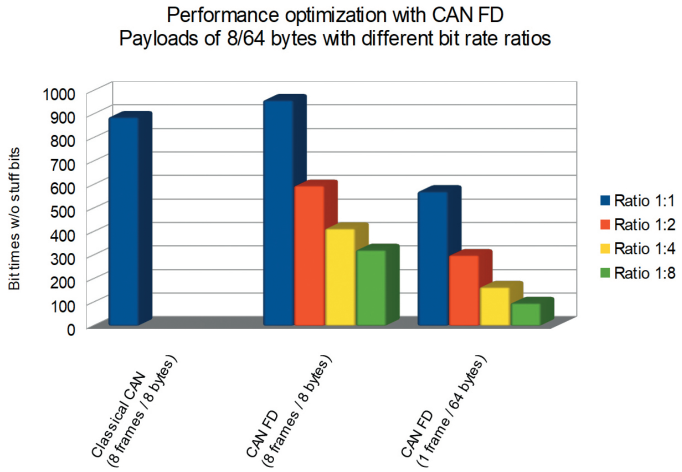


Figure 3: Performance optimization for CAN FD frames with 8-byte and 64-byte payloads sent with different bit rate ratios (Source: ESD Electronics)

With changing the protocol: Even greater performance gains can be achieved if, besides increasing the data rate, the protocol is adapted with additional software effort. In this way the higher number of possible CAN FD data bytes is fully exploited.

Based on the previous example, a protocol (e.g. for firmware update), which was previously based on CAN frames with 8 data bytes, is to be converted to CAN FD frames with 64 data bytes. Figure 3 considers transmission of 64-byte data blocks.

The graph shows that with the protocol unchanged and a ratio of 1:4 between arbitration and data phase bit rate (as in the previous example), the throughput has more than doubled. Using the full 64 bytes of CAN FD user data, the data throughput will increase even fivefold and with a further increase in the bit rate ratio to 1:8 the throughput increase will be more than nine times higher. In the latter case, the transmission of the CAN FD frame with 64 data bytes takes even less time as for transmitting a single CAN frame with 8 data bytes. Under these conditions an additional reduction in latency is achieved.

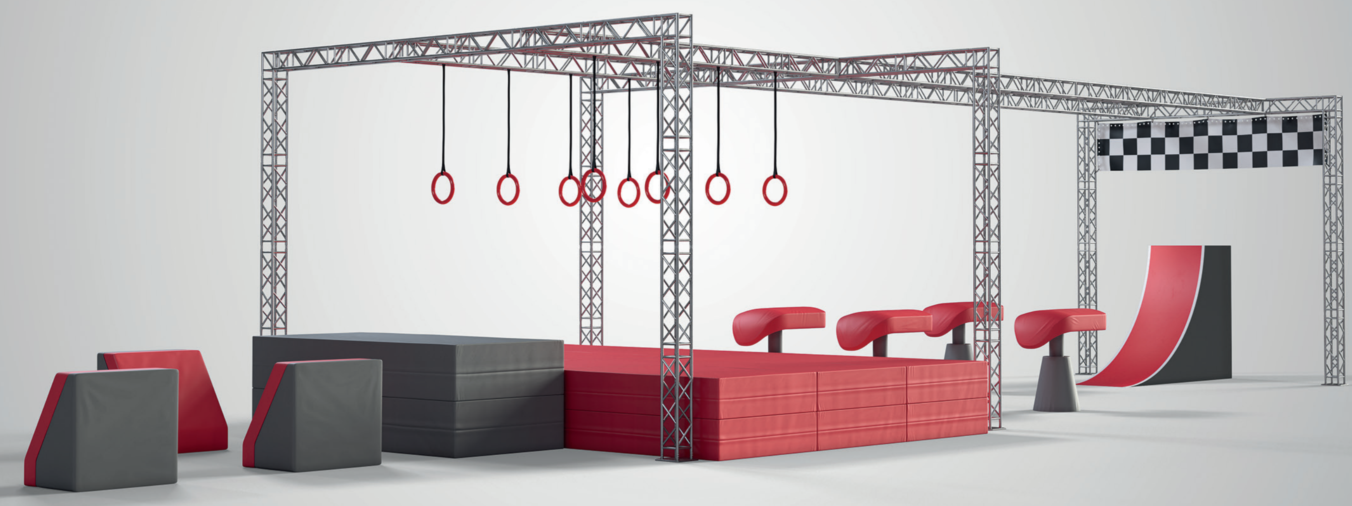
Migration to CAN FD

For hardware developers: For new hardware developments the support of CAN FD has become quite easy. As with Classical CAN, driven by the automotive industry, one or more CAN FD interfaces are replacing the Classical CAN interfaces in current micro-controllers. Up to a certain ratio of the bit rate (arbitration bit rate to data phase bit rate), CAN FD accepts the same oscillator tolerance as CAN, but it is advisable to use a transceiver specified for CAN FD. Even if a customer continues to only use Classical CAN they will benefit from these advantages. If existing designs are to be extended by a CAN FD connection this is easy to implement with now available standalone controllers or by using an FPGA IP core.

For software developers: The effort involved in migrating an application to CAN FD depends heavily on the ▷

Vector Testing Solution

It's Good If Your Software Runs. And Even Better If It Conquers Every Challenge.



Outstanding Software Quality Demands Rigorous Testing.

For embedded software to be successful, it must meet the highest demands. Our comprehensive test solution enables your software project to effortlessly pass every efficiency test. From the design and implementation of the test to its execution, along with the centralized administration and analysis of your test data – we cover the whole gamut, whether for unit or system tests, SIL or HIL. This allows you to always be on the safe side and identify potential improvements for your software at an early stage. And to conquer every challenge with ease.

Vector brings your software development a giant leap forward.

vector.com/testing-solution

VECTOR

Interview: Why migrate to CAN FD?

Dirk Flege (ESD Electronics)

Due to the backward compatibility of CAN FD technology, Classical CAN applications can be migrated to CAN FD or can be used as a basis in new applications. Dirk Flege, Head of Sales at ESD Eelectronics, explained in an interview what makes a migration to CAN FD worthwhile.

Q: The CAN protocol is characterized by great robustness against interference. How can this property be preserved in CAN FD?

A: The CAN FD protocol is designed in such a way that keeping Classical CAN concepts is possible in the arbitration and confirmation phase as well as in error handling. In order to achieve the same robustness against communication errors even in the longer data phase, the protocol uses a 17-bit checksum (frames with up to 16 bytes user data) or a 21-bit checksum (frames with more than 16 bytes user data) instead of the usual 15-bit checksum.

Q: The CAN FD protocol is backwards compatible with the Classical CAN protocol. What are the opportunities of this advantage for industrial automation?

A: Thanks to the backward-compatible design, CAN applications can be easily converted to the more powerful CAN FD communication without having to change the existing wiring. Alternatively, CAN FD components can also be used as a basis in current CAN applications and simply be switched to CAN FD communication later on.

Q: In addition to standardized CAN FD controllers there are also FPGA-based ones available on the market. They have greater flexibility in terms of performance and functional density. What exactly are the advantages of FPGA-based CAN controllers?

A: The write and especially the read access to standard controllers is rather slow compared to the cycle time of modern CPUs (central processing unit). Therefore, we have developed an FPGA-based CAN controller, which all our CAN interfaces are based on. The Advanced CAN Controller (esdACC) has an interface of up to 32 bits wide and supports 64-bit timestamps. Furthermore, it can generate a 100-% busload. These are the features where the CAN FD controller for FPGA is derived from supporting the CAN FD protocol in accordance to ISO11898-1:2015.

Q: What are the specific advantages the FPGA offers for the CAN FD interfaces?

A: The CAN interface "CAN-PCIe/402-FD", for example, is a universal board that was developed for the PCIExpress bus and has one or two CAN FD interfaces in accordance with ISO 11898-2. It uses bus mastering for data transfer to the host storage. This reduces latency during I/O transactions, especially due to the higher data rate and the reduction of CPU load. By using MSI (message signaled interrupts), the PC board can work in hypervisor environments, for example. It also supports high-resolution hardware timestamps.

API (application programming interface) previously used for Classical CAN. If the API remains unchanged for the CAN FD hardware, a migration can take place in three steps.

1. All participants use the CAN FD interfaces as before with the Classical CAN protocol.
2. Conversion of all participants in the data phase with all using a higher bit rate, which with an unchanged protocol will immediately lead to a reduction in latency and bus load respectively to an increase in throughput. The developer must first check whether his protocol requires the sending of RTR frames, since this is no longer supported with CAN FD.
3. Modification/extension of the protocol by transferring more than 8 user data bytes.

In addition to a data throughput gain, the last step facilitates the solution of data consistency problems for transmission of more than 8 user data bytes. Moreover, it is possible to implement protocols, for example in the area of safety and security, which are often difficult or impossible to implement using CAN frames with only 8 user data bytes. Especially in step 3, however, the developer must check whether the desired real-time properties of his implementation (latency times) have still been preserved.

For system integrators: The advantage for system integrators when migrating to CAN FD lies in the fact that the network participants with a Classical CAN controller can initially be exchanged by network participants with a CAN FD controller, even if this takes place in several stages. Due to the backward compatibility, the Classical CAN protocol can still be used for the time being. If there is a need for more bus bandwidth and/or lower latency at a later time, the application can be changed accordingly. Switching back to the Classical CAN state can be done at any time if problems with CAN FD communication arise due to the wiring having been left unchanged. The only limitation to a migration is that a switch to CAN FD can only take place if all network participants support the CAN FD protocol, since Classical CAN controllers will interpret the CAN FD frames as protocol errors.

Higher-layer protocols

After the publication of the standard in 2015, several Classical CAN-based higher layer protocols from different industry sectors were adapted to the CAN FD extensions or are now about to be released. Examples are ISO TP and J1939 (automotive industry), CANopen FD (automation), or Arinc 825 (aviation). ◀

Author

Oliver Thimm
ESD Electronics
info@esd.eu
www.esd.eu

This article was originally published in German language in the magazine "Computer & Automation 8/21".

CAN & CAN FD Connection via Ethernet

PCAN-Ethernet Gateway FD DR

The PCAN-Gateway product family from PEAK-System is designed for the transmission of CAN messages over IP networks. With a single gateway connected to a CAN bus, users can access the CAN bus using the LAN interface of their computer. In addition, different CAN buses can be connected over IP using this technology. The devices are configured via a convenient web interface. Alternatively, the JSON interface allows access via software.

The PCAN-Ethernet Gateway FD DR is the first model supporting the modern standard CAN FD in addition to classic CAN.

Specifications:

- AM5716 Sitara with Arm® Cortex® M15 core
- 2 GByte Flash and 1 GByte DDR3 RAM
- Linux operating system (version 4.19)
- Two High-speed CAN channels (ISO 11898-2)
  - Comply with CAN specifications 2.0 A/B and FD
  - CAN FD bit rates for the data field (64 bytes max.) from 20 kbit/s up to 10 Mbit/s
  - CAN bit rates from 20 kbit/s up to 1 Mbit/s
- Galvanic isolation of the CAN channels up to 500 V against each other, against RS-232, and the power supply
- Connections for CAN, RS-232, and power supply via 4-pole screw-terminal strips (Phoenix)
- LAN interface
  - Data transmission using TCP or UDP
  - 10/100/1000 Mbit/s bit rate
  - RJ-45 connector with status LEDs
- Monitoring and configuration of the devices via the web interface or JSON interface
- Software update via the web interface
- Reboot or reset of the device to a previous software version with a reset button
- Plastic casing (width: 45.2 mm) for mounting on a DIN rail (DIN EN 60715 TH35)
- LEDs for device status and power supply
- Voltage supply from 8 to 30 V
- Operating temperature range from -40 to 70 °C (-40 to 158 °F)

Further PCAN-Gateway Models:

- PCAN-Ethernet Gateway DR - CAN to LAN gateways in DIN rail casing with Phoenix connectors
- PCAN-Wireless Gateway DR - CAN to WLAN gateways in DIN rail casing with Phoenix connectors
- PCAN-Wireless Gateway - CAN to WLAN gateways in casing with flange and D-Sub or Tyco connectors

www.peak-system.com

Take a look at our website for the international sales partners. Scan the QR code on the left to open that page.

PEAK-System Technik GmbH

Otto-Roehm-Str. 69, 64293 Darmstadt, Germany
Phone: +49 6151 8173-20 - Fax: +49 6151 8173-29
E-mail: info@peak-system.com